ANO II
NÚMERO 3 • 1985 • Cr$ 6.500

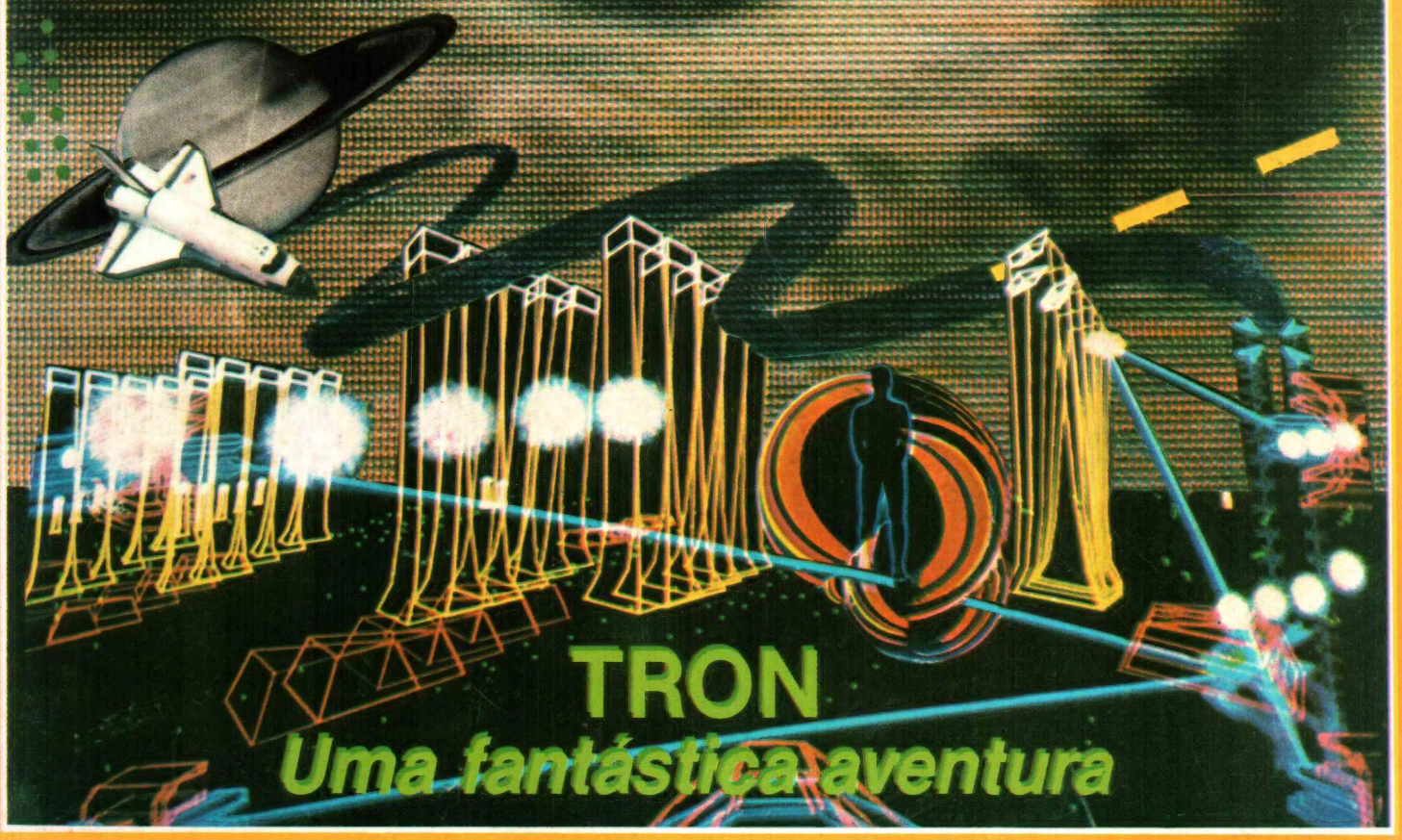
MICROBITS

MACROINFORMAÇÕES E PROGRAMAS PARA MICROS DA LINHA SINCLAIR (TK 82/83/85, CP 200 E COMPATÍVEIS)

A Linha REM

TK na Fórmula 1

Arquivo de Custos



TRON

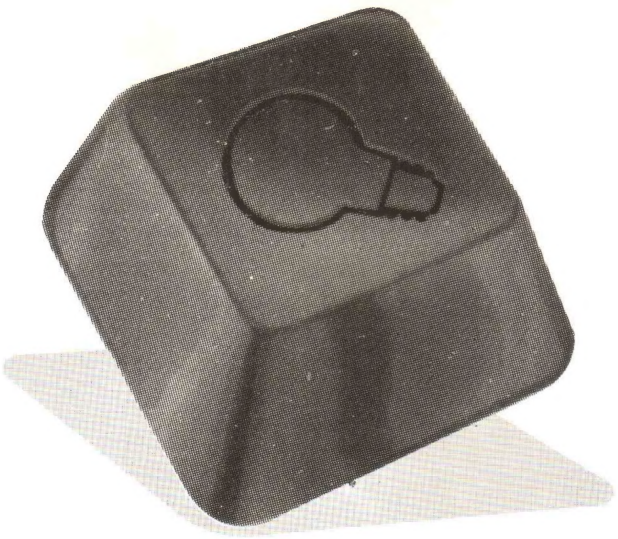
Uma fantástica aventura

Soluções Microidéia

A Microidéia desenvolveu uma linha completa de aplicativos domésticos para as principais famílias de micros fabricados no País. São programas auto-explicativos, com documentação em português, desenvolvidos aqui, e totalmente voltados para as necessidades do usuário brasileiro.

Ignore a pirataria e experimente o software original Microidéia.

Conheça aqui alguns de nossos programas.



O Poder da Simplicidade.

Orçamento Doméstico

Para defender-se da inflação e das leis salariais brasileiras, use o computador. Descubra quais itens estão estourando suas previsões, deixando-o sem dinheiro no final do mês.

CASSETE	Cr$
SO2 Sinclair, Ringo, CP 200 e TK'S	30.000
TO2 TRS 80, CP 500 e 300, JR e DGT 1000	45.000
KO2 TK 2000	45.000
AO2 Apple II	45.000

Diskette	Cr$
DTO2 TRS 80, CP 500 e CP300	220.000
DAO2 Apple II e compatíveis	220.000
DKO2 TK 2000	220.000

Fluxo de Caixa

Controle eficientemente seu fluxo de receitas e despesas, sabendo com antecedência os superavits e deficits de caixa. Com isso você ganha tempo para negociar as melhores taxas de aplicação ou empréstimos.

Cassete	Cr$
SO6	30.000
TO6	45.000
KO6	45.000
AO6	45.000

Diskette	Cr$
DTO6	220.000
DAO6	220.000
DKO6	220.000

Controle de Estoque

Saiba exatamente a quantidade de mercadorias que possui em estoque, evitando que os produtos se esgotem antes de serem repostos. Ponha sua empresa na era da Informática, com o controle compudorizado de estoque.

Cassete	Cr$
SO7	40.000
TO7	45.000
AO7	45.000
KO7	45.000

Diskette	Cr$
DTO7	220.000
DAO7	220.000
DKO7	220.000

Controle Bancário

Se Seu talão de cheques bater com o computador de seu banco utilize seu próprio computador para manter suas contas bancárias sempre atualizadas. Evite os descuidos e nunca você emitirá cheques sem saldo suficiente.

Cassete	Cr$
SO1	30.000
TO1	45.000
KO1	45.000
AO1	45.000

Diskette	Cr$
DTO1	220.000
DAO1	220.000
DKO1	220.000

Contas a Pagar

Controle de maneira eficiente a data de pagamento de suas contas, com totais pagos e a pagar. Listagem em impressora e em tela. Não pague multa!!

Cassete	Cr$
SO8 Sinclair, TK, Ringo e CP 200	30.000
TO8 TRS 80, CP 500, CP 300. JR E DGT 1000	45.000
KO8 TK 2000	45.000
AO8 Apple II e compatíveis	45.000

Diskette	Cr$
DTO8, TRS 80, CP 500, CP 300 e NAJA	220.000
DAO8	220.000
DKO8 TK 2000	220.000

Cadastro de Clientes Mala Direta

Cadastre seus clientes, amigos e fornecedores com dados completos e emita listagens ou etiquetas. Não tendo impressora, veja a listagem na tela.

Cassete	Cr$
SO9 Sinclair, TK, Ringo e CP 200	30.000
TO9, TRS 80, CP 500, CP 300, RJ e DGT 1000	60.000
KO9, TK 2000	45.000
AO9 Apple II e compatíveis	45.000

Diskette	Cr$
DTO9, TRS, CP 500, CP 300 e NAJA	220.000
DAO9 Apple e Compatíveis	220.000
DKO9 TK 2000	220.000

Quero adquirir o(s) seguinte(s) programa(s) Microidéia:

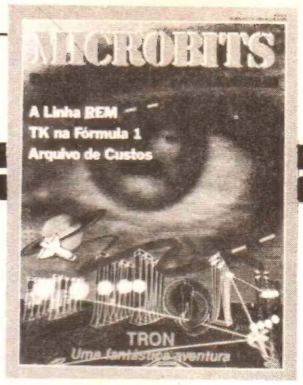
Código	Quant.	Preço	
Total			

Estou enviando um cheque do banco _____ de n.º _____ à Microidéia Ltda.

Microidéia
Caixa Postal 6151
20022 — Rio de Janeiro — RJ

Nome _____
End. _____
Bairro _____ Cidade _____
Tel.: _____ CEP _____
Micro _____

☐ **Envie-me um catálogo de programas Microidéia.**



Diretor Editorial
David Thomas Anderson

Editor Responsável
Sebastião Miranda

Consultor Técnico
Bernhard Wolfgang Schön

Redação
Eni Barreto Anderson

Programação Visual
Otavio Studart

Arte
Jorge Monteiro
Valmir Oliveira

Colaboradores
Nelson N. S. Santos, Alexandre Pfeifer, Rodolfo Eduardo Hradec, Manoel Silva Rodrigues, Marcel G. de Albuquerque, Maurício Costa Reis, Renato Strauss, Sinésio Amorim Filho, Gelson Dias Santos e Eduardo Marchiori

Publicidade
Rio de Janeiro
Editora Campus Ltda.
Rua Barão de Itapagipe 55 Rio Comprido
CEP 20261 RJ — RJ — Tel.: (021) 284-8443
São Paulo
MICRO BITS Informática Ltda.
Cx. Postal 12464 — CEP 04798 SP — SP
Tel.: (011) 521 4235

Circulação e Assinaturas
Editora Campus Ltda.

Distribuição
Fernando Chinaglia
Rua Teodoro da Silva, 907 — RJ Tel.: (021) 268 9112

Impressão e Acabamento
Graphos Ltda.

MICROBITS é uma publicação bimestral da Editora Campus Ltda., Rua Barão de Itapagipe 55 Rio Comprido — RJ CEP 20261 Tel.: (021) 284-8443

© 1985, Editora Campus Ltda.

Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/1973.
Nenhuma parte desta revista poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Todo o esforço foi feito para fornecer a mais completa e adequada informação. Contudo a editora não assume responsabilidades pelo uso da mesma.
A Editora Campus não é filiada a nenhum fabricante de sistemas computacionais.

CORRESPONDÊNCIA

1. Assinaturas
Desejando obter informações sobre assinaturas, escreva para:
"Assinaturas"
Revista MICROBITS
A/c Editora Campus Ltda.
Rua Barão de Itapagipe 55 — 20261 Rio de Janeiro, RJ
Tel.: (021) 284-8443

2. Redação
Envie suas cartas, programas e artigos para:
Ao Diretor Editorial — Revista MICROBITS
Caixa Postal 12464 — 04798 São Paulo, SP.

LEMBRETE IMPORTANTE

Esclarecemos mais uma vez aos nossos leitores que a Editora Campus não é responsável por quaisquer questões relativas aos números de MICROBITS anteriores ao Ano II — nº 1. Assim, solicitamos que as mesmas sejam diretamente tratadas com o Diretor Editorial, Sr. David Anderson, no seguinte endereço: Caixa Postal 12464 — 04798 — São Paulo — SP

Editorial

Nos tempos atuais de crise, somos constantemente lembrados dos altos índices de desemprego no país. Uma exceção parece ocorrer na área da Informática. Um programador ou analista de sistema *com experiência*, normalmente não tem dificuldade em conseguir emprego. Mas e os que não têm a experiência exigida pelas firmas? Ficam na difícil situação de: (1) não conseguir experiência porque não têm emprego, e (2) não conseguir emprego porque não têm experiência!

Como resolver esta situação? Se todos os gerentes de CPD, chefes de setores de informática, e todos os que regularmente procurem mão-de-obra especializada, lembrarem que há muitos excelentes futuros operadores, programadores, analistas de sistema etc., esperando a primeira chance, porque não incluir em seus planos alguns estagiários, programadores trainees,...? O custo será muito baixo e eles provavelmente se mostrarão realmente aptos a prestar serviços úteis logo após sua contratação. Afinal, os gerentes, chefes etc., estiveram na mesma posição no início de suas carreiras — transpondo-se a barreira do primeiro emprego, fica mais fácil conseguir outro.

MICROBITS costuma incluir artigos e programas de leitores que nunca escreveram antes para revistas ou outras publicações — por falta de conhecimento do assunto, tempo, coragem, ou o quer que seja. Recentemente recebemos vários programas de Alexandre Pfeifer de Divinópolis, MG, que, embora tendo somente 14 anos na época em que nos enviou o material, mostra uma habilidade em programar fora do comum. Publicamos o primeiro programa dele na última edição ("FORCA" em linguagem de máquina) e queremos continuar incluindo seu material no futuro. Esperamos estar contribuindo na formação destes iniciantes, dando-lhes o apoio de que provavelmente sentem muita falta. Assim, reiteramos nosso convite aos leitores para que enviem programas e artigos à nossa redação. Qualquer contribuição será analisada e, se considerada de nível adequado, publicada futuramente. Podemos citar ainda nosso amigo Bernhard Wolfgang Schön, que começou escrevendo artigos para MICROBITS em 1983 e hoje é reconhecido como um "expert" em hardware. Talvez haja entre nossos leitores outros "Bernhards" esperando ser "descobertos".

David Thomas Anderson

Mais uma vez estamos aqui com vocês. Trinta e duas páginas quase que exclusivamente de artigos, programas, aplicativos, dicas e comentários para todos os gostos. E por falar em gosto, precisamos saber o *seu* — para poder atendê-lo. Seus pedidos, críticas e sugestões são muito importantes para que possamos voltar ainda melhor no número 4. Até lá.

Editora **CAMPUS**

Índice

Cartas 4
Desafio 5
A Linha REM 6
Torres de Hanói 9
Arquivo de Custos 10
TK na Abertura da Fórmula 1 12
Programa Integral 16
Além do BASIC 18
TRON — Uma Fantástica Aventura 22
Aproximação de Romberg 24
Notícias 25
Programa Ordenador 26
Classificados 28
Dicas 29
Livros 30

Cartas

• Após ler minuciosamente essa incrível revista, observei na página 3, onde estava escrito "correspondência", um pedido de envio de cartas, programas e artigos. Resolvi então mandar um dos inúmeros programas em Assembler que eu (Max Oliver) e meu amigo (Vladimir Lemos) desenvolvemos.

Este programa, o HI-SPEED para TK, CP200 e outros computadores de lógica SINCLAIR, seria uma solução para a transferência de dados em um pequeno espaço de tempo. Enquanto a velocidade normal de gravação e reprodução é de 300 bauds, o de HI-SPEED é de 4200 bauds ou 14 vezes mais rápido do que o normal.

O programa foi inspirado no HI-SPEED do TK85 e num programa vendido pela SOFT-KRISTIAN. Ambos eram muito grandes. Então nós os analisamos e conseguimos este que possui três comandos e ocupa 470 bytes da RAMTOP e não pode ser apagado pelo NEW.

MAX OLIVER E VLADIMIR TORIBIO LEMOS,
Rio de Janeiro — RJ

µBits: *Agradecemos o envio do programa em linguagem de máquina. Após análise, esperamos publicá-lo numa futura edição.*

• Solicito que seja feita uma revisão no programa "DESENHE NA TELA" na página 29 da revista Ano II, nº 1, pois ao pressionar a tecla "A", ao invés de aparecer a "Pena da caneta" aparece o erro "2/100", que indica variável indefinida na linha 100.

DAVID BEN AURAM,
São Paulo — SP

µBits: *Houve uma pequena falha na impressão deste programa. A causa do erro foi na linha 30, que deveria ser:*

30 LET A = 0

Infelizmente a letra "A" saiu como "H" na listagem.

Como a maioria dos programas publicados são testados e impressos numa impressora TS 2040, é muito difícil passar erros. Mesmo assim, sempre há a possibilidade de ocorrer falha. Esperamos que nos sejam notificados quando localizados defeitos deste tipo.

• Primeiro, quero parabenizá-los pela excelente qualidade da revista, principalmente por abranger somente a área dos micros que utilizam a lógica SINCLAIR.

O objetivo desta carta é eliminar certas dúvidas:

— Na Revista Ano II, nº 1, pude constatar no programa "Como Criar Caracteres Especiais no Seu Micro" os seguintes sintomas que presumo estarem errados:

1. Ao digitar o caractere a ser redesenhado pelo computador, este nos fornece uma matriz aleatória que não se parece com nenhum caractere.

2. Depois de fornecer-lhe as coordenadas para mudança do caractere, e ao pedir que o computador demonstre o novo caractere no tamanho original, aparece no lado esquerdo inferior do vídeo a mensagem: "D/460".

Desejaria saber, se possível, como armazenar vários caracteres mudados e depois imprimi-los um ao lado do outro, organizando composições maiores.

Portanto, gostaria que os senhores corrigissem o devido programa e me enviassem a listagem já corrigida.

MARCELO MAGALHÃES MARQUES,
Rio de Janeiro — RJ

µBits: *Para que o referido programa funcione, Marcelo (e outros que nos escreveram com o mesmo pedido), é necessário implementar as modificações de hardware mencionadas em MICROBITS Ano I, nº 6.*

• Tenho 13 anos e gosto muito de computadores. Possuo um TK85 e gostaria de receber informações de como adquirir uma impressora SINCLAIR.

Queria também saber se é possível fazer um jogo com imagem tridimensional com alta resolução no TK, por exemplo "Defender". E, finalmente, gostaria de receber um programa qualquer para sintetizador de som.

EDUARDO SCHNEIDER,
Av. Morumbi, 6800, Morumbi
05650 — São Paulo — SP

µBits: *Não é fácil conseguir uma impressora SINCLAIR aqui no Brasil. Sugerimos que leia os classificados das revistas de microinformática — às vezes aparecem anúncios de venda deste equipamento.*

Antes de desenvolver programas em alta resolução é necessário fazer as modificações no seu micro para adaptá-lo a funcionar daquele modo. Embora seja possível desenvolver jogos tridimensionais, é óbvio que isto seria muito mais difícil do que os jogos bidimensionais. Sugerimos, então, que experimente com programas animados simples e tente sofisticá-los cada vez mais.

Não possuímos informações sobre programas para sintetizador de som. Caso haja leitores com dados sobre este assunto talvez possam entrar em contato diretamente com Eduardo.

• Coleciono a Enciclopédia Prática de Informática e em um dos seus fascículos saiu o jogo "Porquinho" para microcomputadores: CP300, CP500, Sysdata Jr. etc. e eu consegui fazer uma versão para o meu TK85. Gostaria de saber se versões como essa, de outras coleções, podem ser enviadas para serem analisadas por vocês, e se devem ser datilografadas ou gravadas em fita.

MARCELO DE MACEDO,
Rio de Janeiro — RJ

µBits: *Programas adaptados para outros micros não podem ser considerados cópias exatamente, e não deve haver problemas em publicar o seu. Entretanto, preferimos receber programas desenvolvidos pelos nossos leitores, ou aqueles que foram melhorados, e não somente transcritos para outros micros.*

Quanto a maneira de enviar programas para serem publicados, nossa preferência é de recebê-los gravados em fita cassete (ao mínimo duas gravações

na fita), pois assim eliminam-se erros de digitação. Caso isto não seja possível, outra alternativa seria a de enviar os programas datilografados ou manuscritos, tomando muito cuidado quando copiando da tela e, subseqüentemente, fazendo uma verificação linha por linha para conferir.

• Venho por meio desta informar que a dica que lhe enviei e foi publicada no nº 2, apresentou um erro de imprensa na linha 100. O correto é

```
100 LET I$=STR$((P-INT P)+ 0.005)
```

Seria bom que fosse publicada uma errata pois, como está, o programa irá parar na linha 100 com erro 2 (variável não definida).

Também observei que meu nome e cidade saíram errados.

ÁLVARO FERREIRO BORJA,
Belém – PA

µBits: *Desculpe a nossa falha, Álvaro!*

• Adquiri, recentemente, a revista MICROBITS nº 2 (Ano II) e gostaria, em primeiro lugar, de parabenizá-los pelo excelente nível da mesma. Dinâmica, com assuntos abordados de maneira clara e inteligente. Porém surgiu-me uma dúvida quanto ao programa FORCA em linguagem de máquina, pois a respeito da linha REM para acomodar o bloco em Assembler, não há uma indicação de com quantos caracteres deve ser criada esta linha.

LUCIANO PESSOA MENDONÇA,
Natal – RN

µBits: *A linha deve ter 995 caracteres (ao mínimo) após o "REM".*

CHARADA

```
16514  2A 0C 40 44  4D 16  38 CB
16522  22 2A 10 40  0A BA  28 03
16530  C6 80 02 03  BF ED  42 20
16538  F0 C9
```

O que faz este programa em linguagem de máquina? Isto é relativamente fácil de descobrir. Para rodá-lo, digite (modo direto) ou inclua como linha de programa RAND USR 16514.

A charada é: VOCÊ PODE EXPLICAR **COMO** ELE FUNCIONA, INSTRUÇÃO POR INSTRUÇÃO? Faça a desmontagem do programa e PENSE!!!

Escreva-nos com sua resposta, publicaremos os nomes e as soluções dos acertadores.

NELSON SANTOS
Rio de Janeiro – RJ

seção DESAFIO

A excelente revista Micro Sistemas, em seu número 37 (outubro 1984), publicou na seção "Dicas", página 81, a seguinte:

"Ilusão de ótica

Seja enganado pela rapidez de Assembler com esta dica. Primeiro, digite a linha

1 REM ..... (30 caracteres) .... *

Em seguida, entre com os códigos em hexadecimal ...

```
16514  00  2A  0C  40  23  06  16
       0E  21  7E  FE  76  28  04
       3A  82  40  77  23  0D  20
       F3  10  EF  C9
```

Depois de entrar com os códigos, digite este programa em BASIC:

```
 5 POKE 16514,6
10 RAND USR 16515
15 POKE 16514,134
20 RAND USR 16515
25 RUN
```

Rode esta dica e acerte o desafio de tentar descobrir qual o sentido do movimento apresentado na tela.

JEAN MARCELO DE OLIVEIRA – SP

Observe que da maneira como estão combinados a rotina ASSEMBLY e o programa BASIC, a única maneira de interromper o programa é pressionando a tecla BREAK, já que a linha 25 reinicia sempre a execução.

Nosso DESAFIO é o seguinte – escrever uma rotina ASSEMBLY que atenda às seguintes especificações:

* faça o *mesmo* efeito visual;
* *dispense* ser "cercada" por um programa BASIC;
* seja acessada por RAND USR 16514;
* *repita* o efeito *256* vezes;
* *gaste o mesmo número de bytes que a original* :*25* !

Será que está claro?

a) Ø REM 25 caracteres quaisquer;
b) entrada dos novos códigos hexa;
c) a digitação de RAND USR 16514 (modo direto) deve produzir a repetição do efeito 256 vezes!

No próximo número publicaremos o(s) programa(s) acertador(es) que nos chegarem *devidamente explicados*, bem como (naturalmente) o(s) nome(s) de seu(s) autor(es).

Se por acaso não houver acertador algum – no que não acreditamos – publicaremos a solução e ... um novo DESAFIO. Até lá, e boas dores de cabeça.

EM TEMPO: Enviem colaborações para esta seção! Tanto faz em BASIC ou em ASSEMBLY, o importante é a CRIATIVIDADE. Não escondam seus talentos.

* Bem, aqui há certamente um erro de impressão: apenas 25 caracteres são necessários.

Nelson N. S. Santos

A LINHA REM

MISTURE ASSEMBLY COM BASIC

O resultado é maior rapidez e eficiência

Uma das virtudes do programador é saber quando usar o BASIC, e quando usar a linguagem de máquina. Como regra geral, diríamos que o BASIC é mais fácil — devido a ser linguagem de alto nível, e mais lento — devido a ser linguagem interpretada.

Assim, certas instruções em BASIC tornam-se particularmente lentas, em especial loops FOR/NEXT. Logo, é vantagem substituí-los quando possível por loops em linguagem de máquina, apesar da programação ASSEMBLY ser bastante mais trabalhosa.

Toda esta introdução é para falarmos sobre programas para geração de linhas REM com grande número de caracteres, necessárias para programas ASSEMBLY extensos. Estudamos com cuidado diversas matérias sobre o assunto (ver bibliografia no final deste artigo), em especial a de Antonio Sílvio Cezar, publicada em MICROBITS, Ano II, Nº 1, e acreditamos ter criado um programa *híbrido* BASIC e ASSEMBLY bastante prático e rápido.

Sugerimos ao leitor de MICROBITS que *grave* este programa em fita de boa qualidade, pois irá usá-lo muitas e muitas vezes daqui para frente. Entre com as seguintes listagens, nesta ordem. As explicações dos detalhes estão no final do artigo.

```
1 REM 48 caracteres quaisquer
POKE 16510,0 (modo direto)
```

LISTAGEM 1 — Reservando espaço para o programa ASSEMBLY

```
 999 STOP
1000 LET E=16514
1010 LET H$=""
1020 IF H$="" THEN INPUT H$
1030 IF H$="S" THEN STOP
1040 PRINT E,H$( TO 2)
1050 POKE E,16*CODE H$+CODE H$(2)-476
1060 LET E=E+1
1070 LET H$=H$(3 TO )
1080 IF PEEK 16442=2 THEN CLS
1090 GOTO 1020
```

LISTAGEM 2 — O carregador ASSEMBLY

Digite RUN 1000, e dê entrada ao programa ASSEMBLY da listagem 3 (quantos códigos você quiser de cada vez). Após C9 digite S para interromper.

```
16514 CD 23 0F  A  CALL FAST
16517 01 00 00  B  LD BC,0
16520 21 B8 40  C  LD HL,16568
16523 E5        D  PUSH HL
16524 C5        E  PUSH BC
```

```
16525 CD 9E 09  F  CALL MAKE ROOM
16528 C1        G  POP BC
16529 2A B5 40  H  LD HL,(16565)
16532 09        I  ADD HL,BC
16533 22 B5 40  J  LD (16565),HL
16536 2A 29 40  K  LD HL,(NXTLIN)
16539 09        L  ADD HL,BC
16540 22 29 40  M  LD (NXTLIN),HL
16543 E1        N  POP HL
16544 36 17     O  LD (HL),23
16546 23        P  INC HL
16547 0B        Q  DEC BC
16548 79        R  LD A,C
16549 B0        S  OR B
16550 20 F8     T  JR NZ,O
16552 CD 2A 0A  U  CALL CLS
16555 CD 2B 0F  V  CALL SLOW
16558 CD 30 07  W  CALL LIST
16561 C9        X  RET
```

LISTAGEM 3 – O programa ASSEMBLY

NÃO digite RAND USR 16514 para "testar". Tudo que você irá obter é um lindro CRASH. Para terminar digite o programa BASIC da listagem 4.

```
2  REM
10 PRINT "COMPRIMENTO DA REM?"
20 INPUT C
30 POKE 16518,C-256*INT(C/256)
40 POKE 16519,INT(C/256)
50 RAND USR 16514
```

LISTAGEM 4 – Programa BASIC para cercar a rotina ASSEMBLY

Após ter entrado com as quatro listagens, vamos testar o funcionamento do programa. Digite RUN e responda à pergunta da linha 10 com qualquer número pequeno, por exemplo 7. Num piscar de olhos (e num piscar de tela também, pois a rotina em linguagem de máquina entra em FAST), lá está a sua linha 2 REM com 7 asteriscos.

Para fazer novos testes, elimine a linha 2 REM "produzida" (na verdade apenas *expandida* pelo programa) e volte a colocar 2 REM sem nenhum carácter após. O programa é bastante rápido. Não conseguimos cronometrar o tempo de expansão para 1000 bytes – é certamente *menor que um segundo*.

Se for necessário, você pode produzir quantas linhas REM precisar, com quantos caracteres quiser! Suponhamos que você precise de uma linha com 150 e outra com 300 caracteres. Primeiramente faça uma com 150 caracteres. Edite-a, troque seu número para qualquer número livre no programa (por exemplo 5), e substitua a linha 2 com 150 caracteres por novamente 2 REM. Rode outra vez o programa, e crie a linha 2 REM com 300 caracteres. Viu? Fácil e rápido.

Após fazer todos os testes que desejar, *apague* todas as linhas "criadas", volte a colocar 2 REM e *grave* o programa numa fita de boa qualidade, *sem apagar* o carregador ASSEMBLY. Você tem agora uma ferramenta poderosa para auxiliá-lo na entrada de códigos de máquina.

Depois de gravado, e depois de utilizado para a finalidade que você desejou, *apague* a linha 0, e as linhas de 10 até 50 – elas já terão cumprido seu papel.

Agora, vamos às explicações teóricas, analisando em profundidade o programa ASSEMBLY.

A) A rotina trabalha em FAST. O endereço de FAST na ROM é 0F23 (3875). Digitar RAND USR 3875 é equivalente a digitar FAST. Você pode "chamar" FAST em seus programas em linguagem de máquina da mesma forma que fizemos.

Precisamos agora preparar o terreno para a chamada da rotina MAKE ROOM da ROM que expande o programa BASIC. Para tal é necessário que:

- BC contenha o número de bytes de que o programa vai ser expandido. Lembre-se sempre: BC = *b*yte *c*ounter (contador de bytes).
- HL contenha o endereço do byte que será "transferido para frente".

B) Da forma como digitamos o programa, BC contém 0. Vamos colocar em BC o número de bytes desejado através das linhas 30 e 40 do programa BASIC (listagem 4): 30 "faz" o conteúdo de C e 40, o de B.

C) HL é carregado com 16568, que é o endereço do byte que será "transferido para frente", veja: o programa ASSEMBLY vai de 16514 até 16561, logo 16562 contém o NEW LINE de fim de linha, 16563 e 16564 contêm o número da linha 2, 16565 e 16566 contêm o comprimento da linha 2, 16567 contém o código de REM (234), e 16568 contém o primeiro byte da linha 2, que atualmente é um NEW LINE de fim de linha; este NEW LINE será "transferido para frente" de tantos bytes quanto o conteúdo de BC.

D e E) Como a maioria das rotinas da ROM, MAKE ROOM também destrói os registros. Logo, é necessário salvá-los na pilha.

F) Finalmente, a chamada da rotina. Seu endereço é 099E (2462), e ela expande o programa BASIC – conseqüentemente também desloca o arquivo de imagem e a área das variáveis.

G) Recupera o valor de BC da pilha.

H) Carrega HL com o conteúdo dos endereços 16565 e 16566, que contêm o comprimento da linha 2 REM ainda sem nenhum carácter. Este comprimento precisa ser corrigido.

I) Adiciona BC a HL para obter em HL o novo comprimento da linha.

J) Carrega 16565 e 16566 com o novo comprimento.

K) Carrega HL com o conteúdo da variável do sistema NXTLIN, cujo endereço é 4029h (16425d). Ela contém o endereço da próxima linha de BASIC a ser executada. Este endereço precisa ser corrigido.

L) Adiciona BC a HL para obter em HL o novo endereço.

M) Carrega NXTLIN com o novo endereço.

N) Recupera o valor de HL da pilha, ou seja, 16568.

O) Coloca em 16568 o "carácter de construção" da li-

nha REM — escolhemos asterisco, mas você pode alterar isto de acordo com seu gosto.

P) HL aponta para a próxima posição.

Q) BC vai ser usado como contador do loop. Como é possível? Você vai entender: BC foi decrementado, mas *DEC par de registradores não afeta a ZERO FLAG*, logo JR NZ ou JR Z *não* funcionarão! Então, como verificar se BC atingiu zero? Através da instrução lógica (booleana) OR, que *afeta* a ZERO FLAG. Veja como proceder.

R) Carrega o acumulador com o conteúdo de C.

S) Executa entre A e B (ou seja, entre B e C) a operação OR, que funciona — bit a bit — da seguinte maneira:

1 OR 1 = 1
1 OR Ø = 1
Ø OR 1 = 1
Ø OR Ø = Ø

Ou seja, o resultado da operação lógica OR B só será zero quando *todos* os bits de A e de B forem Ø, o que indicará que o conteúdo do par de registradores BC chegou a zero. Bastante engenhoso, e utilíssimo, pois permite loops com um número de voltas maior que 256.

T) Volta ao começo do loop enquanto BC não atingir Ø..

U) Da mesma forma que nas instruções A e F, vamos chamar uma rotina na ROM. O endereço de CLS é ØA2A (26Ø2). Digitar RAND USR 26Ø2 é equivalente a digitar CLS.

V) RAND USR 3883 = SLOW
W) RAND USR 184Ø = LIST
X) RET é obrigatório ...

Finalmente, uma bibliografia. Citamos aqui todos os artigos *brasileiros* que lemos sobre o assunto. Esta citação para nós é motivo de orgulho: há *brasileiros* escrevendo material *de qualidade* sobre praticamente qualquer ramo da microinformática!

ARIGA, Nancy Mitie. A linha REM. In: — Microhobby. Um programa ASSEMBLY de 96 bytes que realmente *gera* uma linha REM (e não apenas a expande, como fizemos).

CEZAR, Antonio Sílvio. Crie uma linha REM do tamanho que você deseja sem digitá-la. In: — MICROBITS, Ano II, nº 1. Já o citamos no início deste artigo.

HENDRICK, Daniel. Um REM de infinitos bytes. In: — Micro Sistemas, nº 30. Um programa ASSEMBLY de 36 bytes que *expande* uma linha REM.

SCHAEFFER, Mario. Usando linguagem de máquina, Editora Moderna, 1984. No apêndice A deste livro há um programa muito semelhante ao de Nancy, citado anteriormente. O programa também ocupa 96 bytes.

Esperamos ter trazido nossa contribuição ao assunto, e colaborado para esclarecer o leitor.

NELSON N. S. SANTOS

Primeiro foi o BASIC BÁSICO
Depois BASIC PARA MICROS
Agora, especialmente para os leitores de MICROBITS:

- **Abordagem simples e objetiva.**
- **Temas apresentados em ordem crescente de dificuldade.**
- **Numerosos exemplos, dicas e problemas.**
- **Seção "Desafio". Ao final dos Capítulos um programa sempre um pouco mais complexo que o nível do capítulo a que pertence.**
- **Apêndices úteis trazendo resumo de todas as funções e comandos dos micros da Linha Sinclair; roteiro para entender e contornar erros de sintaxe; sugestão de folha de codificação que facilita a elaboração de programas, o uso da letra e a exposição dos resultados; uma explicação dos principais termos e nomes técnicos existentes na área e de uso corrente nas revistas especializadas.**

BASIC SINCLAIR

MAIS UM LANÇAMENTO DA EDITORA CAMPUS LTDA.

TORRES DE HANÓI

```
  0 ********************************
    *  TORRES  DE  HANOI  1984     *
    *                              *
    *ALEXANDRE PFEIFER             *
    *CX. POSTAL 242                *
    *35500 DIVINOPOLIS MG          *
    *FONE (037)221-5836            *
    ********************************
  1 SAVE "T"
  2 LET C=0
  5 DIM A$(6,6)
  7 DIM B$(6,6)
  8 DIM C$(6,6)
 10 LET A$(6)="   ▀   "
 20 LET A$(5)="  ▄▀▄  "
 30 LET A$(4)=" ▄▀▄▀▄ "
 40 LET A$(3)="▄▀▄▀▄▀▄"
 50 LET A$(2)="▀▀▀▀▀▀▀"
 55 LET A$(1)="▄▄▄▄▄▄▄"
 60 PRINT AT 0,0;"          *TORRE DE HANOI*          "
 70 PRINT AT 7,0;"     1          2          3     "
 71 LET X=0
 72 LET D=0
 73 LET E=0
 74 LET F=0
 76 LET R=0
 80 FOR N=1 TO 6
 90 PRINT AT 7-N,2;A$(N);AT 7-N,13;B$(N);AT 7-N,24;C$(N)
100 NEXT N
101 FOR N=1 TO 6
102 IF C$(N)<>"       " THEN LET X=X+1
103 NEXT N
105 IF X=6 THEN GOTO 300
110 PRINT AT 8,0;"DA TORRE ";
120 INPUT D$
130 IF LEN D$<>1 OR D$<"1" OR D$>"3" THEN GOTO 120
140 LET A=VAL D$
150 PRINT CHR$ (CODE D$+128)
160 PRINT AT 9,0;"PARA TORRE ";
170 INPUT D$
180 IF LEN D$<>1 OR D$<"1" OR D$>"3" THEN GOTO 170
190 LET B=VAL D$
195 IF B=A THEN GOTO 170
200 PRINT CHR$ (CODE D$+128)
201 GOSUB 205
203 GOTO 215
205 PRINT AT 21,0;"  <APERTE UMA TECLA QUALQUER.>  "
210 IF INKEY$="" THEN GOTO 210
212 PRINT AT 21,0;"                              "
213 RETURN
215 PRINT AT 8,0;"                ";AT 9,0;"                "
225 LET C=C+1
230 GOSUB 2000+1000*A
280 GOTO 71
300 LET D$=STR$ C
310 FOR N=1 TO LEN D$
320 LET D$(N)=CHR$ (CODE D$(N)+128)
330 NEXT N
340 PRINT AT 9,0;"TRANSFERIU AS ARGOLAS DA TORRE "1"PARA A TORRE "3" EM "D$;" MOVS."
350 PRINT AT 21,0;"QUER JOGAR DE NOVO? (S=SIM N=NAO)"
360 IF INKEY$<>"S" AND INKEY$<>"N" THEN GOTO 360
370 IF INKEY$="N" THEN STOP
380 CLS
390 RUN 2
3000 FOR I=6 TO 1 STEP -1
3100 IF A$(I)<>"       " THEN GOTO 3300
3200 NEXT I
3250 PRINT AT 10,0;"ESTE LUGAR ESTA VAZIO."
3260 GOSUB 205
3270 PRINT AT 10,0;"                    "
3280 RETURN
3300 IF B=2 THEN LET R=1
3320 IF R=1 THEN IF B$(1)="  " THEN GOTO 3450
3330 IF R=0 THEN IF C$(1)="  " THEN GOTO 3450
3340 FOR N=5 TO 1 STEP -1
3360 IF R=1 THEN IF B$(N)<>"  " AND B$(N+1)="       " THEN GOTO 3470
3370 IF R=0 THEN IF C$(N)<>"  " AND C$(N+1)="       " THEN GOTO 3470
3390 NEXT N
3450 LET N=0
3470 FOR Z=1 TO 6
3500 IF A$(I,Z)="▀" THEN LET D=D+2
3505 IF A$(I,Z)="▄" OR A$(I,Z)="▀" THEN LET D=D+1
3510 IF R=1 THEN IF N<>0 THEN IF B$(N,Z)="▄" OR B$(N,Z)="▀" THEN LET F=F+1
3512 IF R=1 THEN IF N<>0 THEN IF B$(N,Z)="▀" THEN LET F=F+2
3515 IF R=0 THEN IF N<>0 THEN IF C$(N,Z)="▄" OR C$(N,Z)="▀" THEN LET E=E+1
3517 IF R=0 THEN IF N<>0 THEN IF C$(N,Z)="▀" THEN LET E=E+2
3520 NEXT Z
3530 IF R=1 THEN IF N<>0 AND D>F THEN GOTO 3580
3535 IF R=0 THEN IF N<>0 AND D>E THEN GOTO 3580
3540 IF R=1 THEN LET B$(N+1)=A$(I)
3545 IF R=0 THEN LET C$(N+1)=A$(I)
3550 LET A$(I)="       "
3570 RETURN
3590 PRINT AT 10,0;"TRANSFERENCIA ILEGAL."
3600 GOTO 3260
4000 FOR I=6 TO 1 STEP -1
4100 IF B$(I)<>"       " THEN GOTO 4300
4200 NEXT I
4250 GOTO 3250
4300 IF B=1 THEN LET R=1
4320 IF R=1 THEN IF A$(1)="  " THEN GOTO 4450
4330 IF R=0 THEN IF C$(1)="  " THEN GOTO 4450
4340 FOR N=5 TO 1 STEP -1
4360 IF R=1 THEN IF A$(N)<>"  " AND A$(N+1)="       " THEN GOTO 4470
4370 IF R=0 THEN IF C$(N)<>"  " AND C$(N+1)="       " THEN GOTO 4470
4390 NEXT N
4450 LET N=0
4470 FOR Z=1 TO 6
4500 IF B$(I,Z)="▀" THEN LET D=D+2
4505 IF B$(I,Z)="▄" OR B$(I,Z)="▀" THEN LET D=D+1
4510 IF R=1 THEN IF N<>0 THEN IF A$(N,Z)="▄" OR A$(N,Z)="▀" THEN LET F=F+1
4512 IF R=1 THEN IF N<>0 THEN IF A$(N,Z)="▀" THEN LET F=F+2
4515 IF R=0 THEN IF N<>0 THEN IF C$(N,Z)="▄" OR C$(N,Z)="▀" THEN LET E=E+1
4517 IF R=0 THEN IF N<>0 THEN IF C$(N,Z)="▀" THEN LET E=E+2
4520 NEXT Z
4530 IF R=1 THEN IF N<>0 AND D>F THEN GOTO 3580
4535 IF R=0 THEN IF N<>0 AND D>E THEN GOTO 3580
4540 IF R=1 THEN LET A$(N+1)=B$(I)
4545 IF R=0 THEN LET C$(N+1)=B$(I)
4550 LET B$(I)="
4570 RETURN
5000 FOR I=6 TO 1 STEP -1
5100 IF C$(I)<>"       " THEN GOTO 5300
5200 NEXT I
5250 GOTO 3250
5300 IF B=1 THEN LET R=1
5320 IF R=1 THEN IF A$(1)="  " THEN GOTO 5450
5330 IF R=0 THEN IF B$(1)="  " THEN GOTO 5450
5340 FOR N=5 TO 1 STEP -1
5360 IF R=1 THEN IF A$(N)<>"  " AND A$(N+1)="       " THEN GOTO 5470
5370 IF R=0 THEN IF B$(N)<>"  " AND B$(N+1)="       " THEN GOTO 5470
5390 NEXT N
5450 LET N=0
5470 FOR Z=1 TO 6
5500 IF C$(I,Z)="▀" THEN LET D=D+2
5505 IF C$(I,Z)="▄" OR C$(I,Z)="▀" THEN LET D=D+1
5510 IF R=1 THEN IF NK>0 THEN IF A$(N,Z)="▄" OR A$(N,Z)="▀" THEN LET F=F+1
5512 IF R=1 THEN IF N<>0 THEN IF A$(N,Z)="▀" THEN LET F=F+2
5515 IF R=0 THEN IF N<>0 THEN IF B$(N,Z)="▄" OR B$(N,Z)="▀" THEN LET E=E+1
5517 IF R=0 THEN IF N<>0 THEN IF B$(N,Z)="▀" THEN LET E=E+2
5520 NEXT Z
5530 IF R=1 THEN IF N<>0 AND D>F THEN GOTO 3580
5535 IF R=0 THEN IF N<>0 AND D>E THEN GOTO 3580
5540 IF R=1 THEN LET A$(N+1)=C$(I)
5545 IF R=0 THEN LET B$(N+1)=C$(I)
5550 LET C$(I)="
5570 RETURN
```

O objetivo deste jogo é transferir as argolas da torre 1 (veja Figura 1) para a torre 3, com um número mínimo de movimentos possíveis. Para isto há a torre 2 como auxiliar. O que torna este procedimento um pouco mais difícil é que nenhuma argola poderá ficar sobre outra menor!

A Figura 2 mostra as posições das argolas num determinado instante do jogo.

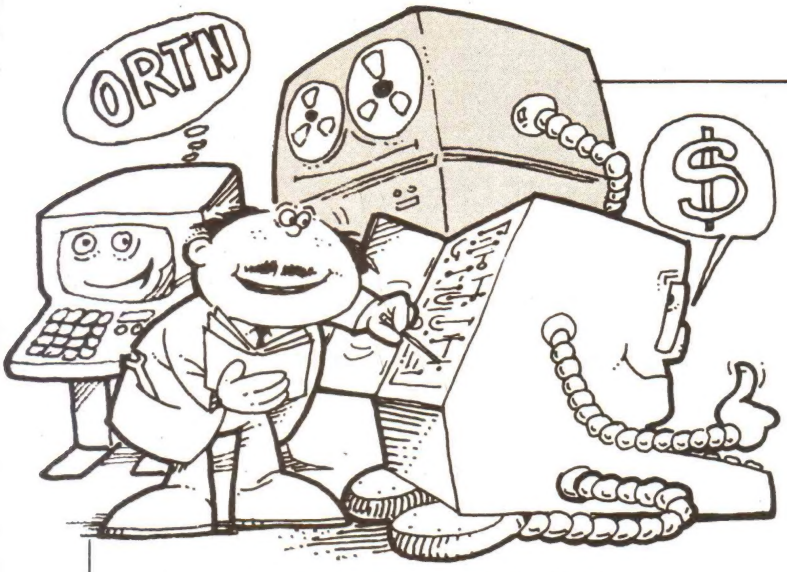
```
          *TORRE DE HANOI*
                ▄
               ▄▀▄
              ▄▀▄▀▄
             ▄▀▄▀▄▀▄
            ▀▀▀▀▀▀▀▀▀
     1          2          3
DA TORRE
```

Figura 1

```
          *TORRE DE HANOI*


                ▄
               ▄▀▄
     1          2          3
DA TORRE 2
PARA TORRE 3
```

Figura 2

ALEXANDRE PFEIFER



ARQUIVO DE CUSTOS

RODOLFO EDUARDO HRADEC

O programa, que é auto-explicativo, calcula o preço de custo, preço de venda e total de custo e de venda de um produto.

Por exemplo, suponha que o produto em foco consiste de vários itens — parafuso, porca, eixo etc. O preço de cada item é digitado junto com a descrição dele e convertido imediatamente para um percentual de ORTNs. Todos os dados relativos a este produto podem ser arquivados em fita para futura modificação. Quando carregado novamente, simplesmente modifique o valor da ORTN para obter preços atualizados.

Um exemplo do seu uso é o seguinte:

Suponha que o produto é fabricado utilizando-se os seguintes materiais:

Item	*Preço*
PORCAS	Cr$ 100
PARAFUSOS	Cr$ 50
CHAPA DE METAL	Cr$ 2500
GABINETE	Cr$ 4000

e a ORTN base é Cr$ 4000.

Também queremos uma margem de lucro de 60%.

Ao rodar o programa (pelo comando direto RUN ou quando carregado da fita) veremos o "menu" abaixo na tela:

```
-----------------------------
    ARQUIVO DE CUSTOS
-----------------------------
VALOR DA ORTN = 4000
INDICE DE LUCRO = 60 %
-----------------------------
   P/ NOVOS ITENS DIGITE <0>
   P/ MUDAR ORTN DIGITE  <1>
   P/ LISTAGEM DIGITE    <2>
   P/ GRAVAR DIGITE      <3>
   P/ MUDAR LUCRO DIGITE <4>
-----------------------------
ARQUIVO COM 4 ITENS

DIGITE O N  E ENTER
```

Se desejarmos recomeçar com um arquivo novo, digitamos "Ø" e NEW LINE (ou ENTER). Depois de digitar o número de itens, temos que entrar com uma descrição de cada item e seu preço, baseado no valor da ORTN armazenada na memória. Obviamente seria necessário modificar a ORTN para o valor atual antes de começarmos a entrar com estes dados, caso não coincida com aquele da memória.

Ao completar a entrada dos dados, podemos modificar a margem de lucro, digitando "4" em resposta ao "menu". Teremos na tela o seguinte:

```
DIGITE O PERCENTUAL DE LUCRO
EXEMPLO   25=25 % DE ACRESC.


O PERCENTUAL ESTA IGUAL A  60
P/ MODIFICAR DIGITE <S>
```

Escolhendo 2 no "menu", aparecerá uma listagem dos itens, com seus preços atuais de custo e venda:

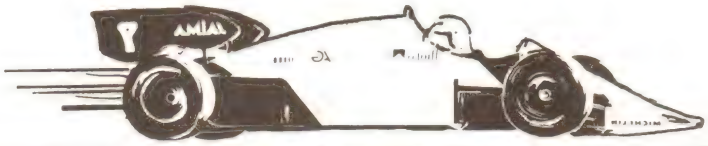
```
DESCRICAO   CUSTO=CR$ VENDA=CR$
-----------------------------
PORCAS        100     160
-----------------------------
PARAFUSOS      50      80
-----------------------------
CHAPA DE METAL 2500    4000
-----------------------------
GABINETE       2000    3200
-----------------------------

PRESS. ENTER
```

e digitando NEWLINE (ENTER) os preços totais de custo e venda:

```
PRECO DE CUSTO = 4650
PRECO DE VENDA = 7440
PRESS. ENTER
```

```
  10 REM ARQUIVO DE CUSTOS
  20 REM RODOLFO EDUARDO HRADEC
  30 LET J=4000
  40 LET Z=0
  50 LET N=0
  60 LET U=1
  70 LET O=710
  80 GOTO 420
  90 PRINT AT 0,5;"ARQUIVO DE CU
STOS"
 100 GOSUB 0
 110 PRINT AT 2,5;"QUANTOS ITENS
?"
 120 INPUT N
 130 PRINT N
 140 GOSUB 0
 150 PRINT AT 10,0;"CUIDADO - IT
ENS SO COM 16 LETRAS"
 160 GOSUB 0
 170 PRINT AT 20,5;"PRESS. ENTER
"
 180 INPUT U$
 190 DIM A$(N,16)
 200 DIM A(N)
 210 CLS
 220 FOR X=1 TO N
 230 PRINT AT 0,0;"CUIDADO - SOM
ENTE 16 LETRAS, O.K."
 240 PRINT ;;"QUAL O ";X;"º ITE
M"
 250 INPUT A$(X)
 260 PRINT A$(X)
 270 PRINT "QUAL O PRECO? (ORTN=
";J;")"
 280 INPUT CUSTO
 290 PRINT CUSTO
 300 LET A(X)=J/CUSTO
 310 PRINT AT 21,5;"PRESS. ENTER
"
 320 INPUT U$
 330 CLS
 340 NEXT X
 350 GOTO 420
 360 PRINT AT 10,0;"DIGITE O PER
CENTUAL DE LUCRO"
 370 PRINT AT 12,0;"EXEMPLO : 25
=25% DE ACRESCO."
 380 GOSUB 1030
 390 PRINT AT 20,0;"DIGITE O PER
CENTUAL DE LUCRO".
 400 INPUT Z
 410 LET U=1+Z/100
 420 CLS
 430 GOSUB 0
 440 PRINT AT 1,4;"ARQUIVO DE CU
STOS"
 450 GOSUB 0
 460 PRINT "VALOR DA ORTN = ";J
 470 PRINT ;;"INDICE DE LUCRO =
";Z;"%"
 480 GOSUB 0
 490 PRINT AT 7,3;"P/ NOVOS ITEN
S DIGITE <0>"
 500 PRINT AT 9,3;"P/ MUDAR ORTN
 DIGITE  <1>"
 510 PRINT AT 11,3;"P/ LISTAGEM
DIGITE     <2>"
 520 PRINT AT 13,3;"P/ GRAVAR DI
GITE       <3>"
 530 PRINT AT 15,3;"P/ MUDAR LUC
RO DIGITE <4>"
 540 GOSUB 0
 550 PRINT "ARQUIVO COM ";N;" IT
ENS"
 560 PRINT AT 20,0;"DIGITE O Nº
E ENTER"
 570 INPUT P
 580 IF P<0 OR P>4 THEN GOTO 570
 590 CLS
 600 IF P=0 THEN GOTO 90
 610 IF P=1 THEN GOTO 650
 620 IF P=2 THEN GOTO 730
 630 IF P=3 THEN GOTO 1010
 640 IF P=4 THEN GOTO 360
 650 PRINT AT 5,4;"DIGITE O VALO
R DA ORTN"
 660 INPUT J
 670 PRINT AT 7,7;J
 680 PRINT AT 20,5;"PRESS. ENTER
"
 690 INPUT U$
 700 GOTO 420
 710 PRINT "--------------------
------------"
 720 RETURN
 730 GOTO 770
 740 PRINT AT 0,0;"DESCRICAO
CUSTO=SOMA VENDA=SOMA"
 750 GOSUB 0
 760 RETURN
 770 GOSUB 740
 780 LET T=0
 790 FOR X=1 TO N
 800 LET T=T+INT (J/A(X))
 810 PRINT A$(X);"  ";INT (J/A(X
));TAB 24;"  ";INT (J/A(X)*U)
 820 GOSUB 0
 830 IF (X/9)=INT (X/9) THEN GOS
UB 950
 840 NEXT X
 850 PRINT ;;"PRESS. ENTER"
 860 INPUT U$
 870 CLS
 880 PRINT "PRECO DE CUSTO = ";T
 890 LET M=INT (U*T)
 900 PRINT ;;"PRECO DE VENDA = "
;M
 910 PRINT ;;"PRESS. ENTER"
 920 INPUT U$
 930 CLS
 940 GOTO 420
 950 PRINT ;;"P/ CONTINUAR PRESS
. ENTER"
 960 INPUT U$
 970 CLS
 980 GOSUB 740
 990 RETURN
1000 GOTO 420
1010 SAVE "ARQUIVO DE CUSTO"
1020 GOTO 420
1030 PRINT AT 18,0;"O PERCENTUAL
 ESTA IGUAL A : ";Z
1040 PRINT AT 20,0;"P/ MODIFICAR
 DIGITE <3>"
1050 INPUT U$
1060 IF U$="3" THEN RETURN
1070 GOTO 420
```

TK



NA ABERTURA DA FÓRMULA 1

No autódromo do Rio, em Jacarepaguá, colaborando com a equipe de cronometragem oficial e junto a sofisticados equipamentos da Longines, Olivetti e outros, estava o pequeno TK, fazendo o controle da corrida. Dando a qualquer momento a posição do carro "X" ou listando todos, mostrando suas posições. Para isto, bastava que lhe informassem a posição relativa de cada carro em cada volta.

Este programa, acima mencionado, criei em meu TK-82C, com teclado da SPEED, chave inversora de vídeo e expansão de 16K. Denominei-o "CONTROLE DE CORRIDAS", pois é para isto que ele se presta.

Grande parte dele é escrita em BASIC e uma pequena, mas a mais importante, em ASSEMBLER. Cabe ressaltar que esta rotina só foi incluída em virtude de que a mesma, escrita em BASIC, levava 14,5 minutos para rodar, tornando o programa impraticável. Em ASSEMBLER passou a levar 3 segundos para 25 veículos.

O programa consiste de:

1 — Inicialmente é perguntado o tipo de veículo que participará da corrida (carro, moto, lancha, bicicleta etc.) e, em seguida, a quantidade de veículos.

2 — A seguir, o programa formata uma tela para entrada de dados (número dos veículos) e a guarda num ponto da memória para não ter que reformatá-la toda vez que se optar por "entrada de dados".

3 — É exibido, então, um "menu" com as seguintes opções:
   E — entrar número dos veículos
   L — lista todos os veículos
   P — lista o veículo "X"
   C — inicializa a matriz de dados
   S — salva o programa (e dados)
   No "menu", é informado também o número da volta em que se encontra a corrida (Figura 1).

4 — A opção "E", exibe uma tela para a entrada dos números dos veículos de acordo com suas posições naquela volta. Quem está em 1º lugar, em 2º, em 3º, etc...
   Na largada (volta zero) devem ser informados todos os veículos participantes. Nas demais, isto não é necessário, pois alguns veículos, por diversas razões, podem ter abandonado a corrida.
   Para o TK saber que aquele foi o último veículo que passou, é previsto o pressionamento da tecla "F" (fim).
   Para eventuais correções de algum número digitado erradamente, pressionando-se "C" o cursor retorna, apagando o número errado, e fica esperando a nova informação.
   Após todos os números terem sido digitados, ainda há a opção para se desprezar tudo que foi digitado. É feita a pergunta: "Tudo correto? (S/N)".
   Tanto para as respostas das perguntas como para a entrada de dados (número dos veículos), não é necessário pressionar "new line" (Figura 2).

5 — A resposta sendo "S" (sim), é processada então a rotina principal em ASSEMBLER que aloca os veícu-

```
          VOLTA 4
    CONTROLE DE CORRIDAS
MENU

  E - ENTRAR COM OS NºS
  P - POSICAO DO VEICULO "X"
  L - LISTA TODOS OS VEICULOS
  C - INICIALIZA MATRIZ
  S - SALVA PROGRAMA

          ESCOLHA
```

FIGURA 1

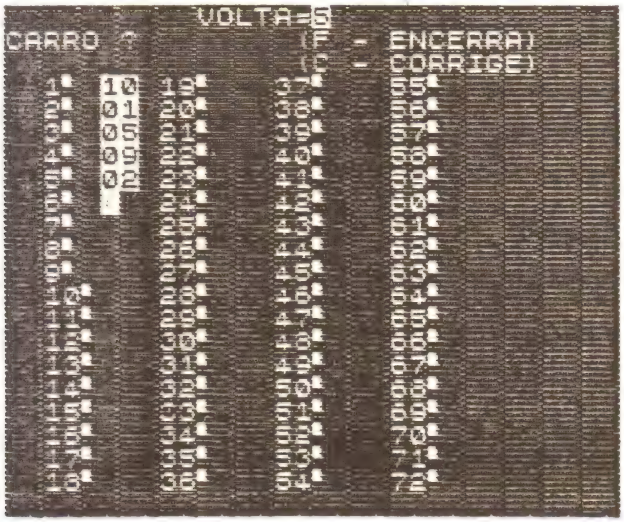


FIGURA 2

los em suas posições relativas e efetivas, considerando a "volta corrente" e a posição da "volta anterior". O processamento volta ao "menu", mostrando o número da próxima volta.

6 – Na opção "L", tem-se a listagem de todos os veículos – quem está em 1º lugar (o número do veículo) e seu número de voltas realizadas; 2º lugar; 3º lugar etc. até o último participante.
É um pouco lenta esta rotina, pois está em BASIC. Em princípio, esta opção só deve ser solicitada ao final da corrida. Pressionando-se qualquer tecla (menos break) retorna-se ao "menu" (Figura 3).

7 – Para se ter uma resposta mais rápida, a opção é "P" – posição do veículo "X". Esta opção, logicamente, pede o número do veículo que se deseja saber a situação (posição e número de voltas). Aqui também é mostrado o número da volta em que se encontra. Com estas informações exibidas, é mostrado também que qualquer tecla retorna ao "menu" (Figura 4).

FIGURA 3

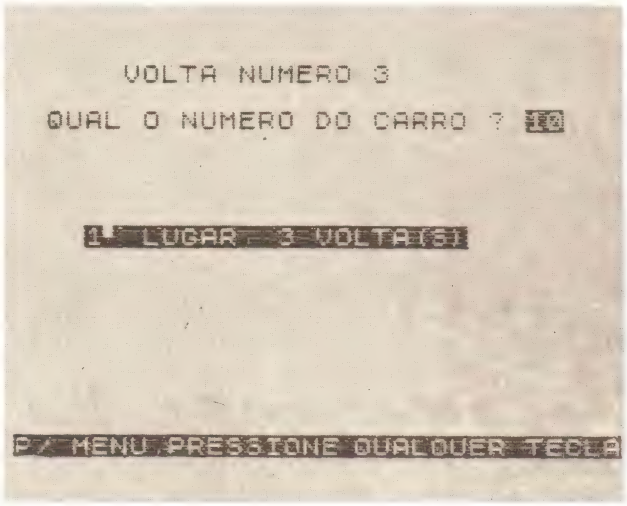


FIGURA 4

8 – A opção "C" deve ser solicitada logo após informar o tipo e o número de veículos, isto é, logo no início. O que ela faz é zerar toda a área de dados e o número de voltas, fazendo exibir, no lugar do número de voltas, a palavra "largada".

9 – Como não podia deixar de existir, pressionando-se "S" é dada a orientação para ligar o gravador e pressionar qualquer tecla. Então, programa e dados daquele momento são gravados no cassete. O programa retorna ao "menu"!

Notas sobre o programa

Para começar, digite uma linha "1 REM..." com, *no mínimo*, 389 pontos – assim:

1 REM ..... (389 pontos).

Pode ser utilizado o programa "Crie uma linha REM" apresentado em MICROBITS, Ano II, nº 1, para facilitar esta tarefa.

Em seguida, digite o programa "carrega". Rode o programinha ("RUN") e carregue os códigos hexadecimais verificando, cuidadosamente, se não há nenhum erro. Apague as linhas 10 a 70 (não apague a linha 1!) e, a seguir, entre com o restante do programa a partir da linha 1000.

Antes de rodar o programa completo, salve-o em fita cassete. Isto evita a perda total do programa, caso ainda haja algum erro na rotina em linguagem de máquina.

```
1 REM .....
.....
.....
.....
.....
```

TK NA ABERTURA DA FÓRMULA 1

```
.....
.....ABS
 SRND0 7( CLS TAN  E£RND7-;S
??$4 POKE ( LIST TAN E£RND)/?;
 GOSUB KTAN  5/? GOSUB ?£RND;
GOSUB KTAN  <>5RND-  CLEAR 5RN
D;/ 2 ?LEN  RETURN $C??UC /  CLE
AR 7 CLEAR 7 CLEAR 7/ FOR <>K I
NPUT ?LEN ?/ CONT <>> RETURN  C
>= CLEAR <>V S LLIST /EXP <>LEN
? RETURN £CAT /SGN ?LEN  <>? ?LE
N  RETURN $C<>7<>7<>7?/TAN
1000 CLS
1020 LET X$="16514+(C-1)*3"
1040 PRINT AT 5,0;"A CORRIDA E
COM QUE VEICULO?"
1060 INPUT F$
1080 PRINT AT 7,0;"QUANTOS(AS) "
;F$;"(S) TERA A","CORRIDA ?";AT
20,3;"(1983) MANOEL ART/SOFT"
1100 INPUT F
1120 CLS
1140 FAST
1160 LET V=0
1180 LET P=V
1200 DIM C$(2)
1220 DIM N$(F,2)
1240 POKE 16806,(F+1)
1260 POKE 16875,F
1280 POKE 16890,(F+1)
1300 POKE 16418,V
1320 PRINT ;;F$;" ?";TAB 15;"(F
- ENCERRA)";TAB 15;"(C - CORRIGE
)"
1340 FOR N=1 TO 18
1360 PRINT TAB 2;N;" ";TAB 8;N+1
8;" ";TAB 14;N+36;" ";TAB 20;N+5
4;" "
1380 NEXT N
1400 PRINT ;;"      (1983) MANOEL
 ART/SOFT      "
1420 RAND USR 16737
1440 RAND USR 16761
1460 CLS
1480 POKE 16418,0
1500 PRINT AT 4,6;"CONTROLE DE C
ORRIDAS"
1520 PRINT AT 6,3;"MENU"
1540 PRINT AT 7,3;"      "
1560 PRINT AT 9,4;"E - ENTRAR CO
M OS N S"
1580 PRINT AT 11,4;"P - POSICAO
DO VEICULO ""X"""
1600 PRINT AT 13,4;"L - LISTA TO
DOS OS VEICULOS"
1620 PRINT AT 15,4;"C - INICIALI
ZA MATRIZ"
1640 PRINT AT 17,4;"S - SALVA PR
OGRAMA"
1660 PRINT AT 21,0;"
                    "
1680 PRINT "      (1983) MANOEL
ART/SOFT      "
1700 RAND USR 16737
1720 SLOW
1740 IF V>0 THEN GOTO 1780
1750 PRINT AT 2,10;"LARGADA"
1760 GOTO 1800
1780 PRINT AT 2,10;"VOLTA";V
1800 PRINT AT 9,4;"E";AT 9,4;"E"
;AT 11,4;"P";AT 11,4;"P";AT 13,4
;"L";AT 13,4;"L";AT 15,4;"C";AT
15,4;"C";AT 17,4;"S";AT 17,4;"S"
;AT 20,10;"ESCOLHA";AT 20,10;"ES
COLHA"
1820 IF INKEY$="" THEN GOTO 1800
1830 LET U$=INKEY$
1840 IF U$="S" THEN GOTO 8000
1850 IF U$="P" THEN GOTO 3000
1860 IF U$="C" THEN GOTO 6000
1870 IF U$="L" THEN GOTO 5000
1880 IF U$<>"E" THEN GOTO 1800
1920 CLS
1940 LET P=0
1960 LET N=P
1980 POKE 16418,P
2000 RAND USR 16774
2020 FOR C=2 TO 27 STEP 6
2040 FOR L=3 TO 20
2060 LET N=N+1
2080 IF N>F THEN GOTO 2400
2100 FOR X=1 TO 2
2120 SLOW
2140 IF V=0 THEN PRINT AT 0,10;"
LARGADA";AT 0,10;"LARGADA"
2160 PRINT AT L,C+2+X;" ";AT L,C
+2+X;" ";AT 0,10;"VOLTA=";V;AT 0
,10;"VOLTA="
2180 IF INKEY$="" THEN GOTO 2140
2200 LET U$=INKEY$
2220 IF U$="C" THEN GOTO 4120
2240 IF U$="F" THEN GOTO 2400
2260 IF U$<"0" OR U$>"9" THEN GO
TO 2140
2280 PRINT AT L,C+2+X;U$
2300 LET C$(X TO X)=U$
2320 NEXT X
2340 LET N$(N)=C$
2360 NEXT L
2380 NEXT C
2400 SLOW
2420 PRINT AT 21,5;"TUDO CERTO ?
 (S/N)";AT 21,5;"TUDO CERTO ? (S
/N)"
2440 IF INKEY$="N" THEN GOTO 192
0
2460 IF NOT INKEY$="S" THEN GOTO
 2420
2480 FAST
2500 IF V=0 THEN GOTO 2660
2520 FOR C=1 TO F
2540 FOR A=1 TO N-1
2560 IF C=VAL N$(A) THEN GOTO 26
40
2580 NEXT A
2600 LET B=VAL X$
2620 POKE (B+1),0
2640 NEXT C
2660 FOR A=1 TO N-1
```

```
2680 LET C=VAL N$(A)
2700 LET B=VAL X$
2720 IF V=0 THEN GOTO 2780
2740 LET C=PEEK (B)+1
2760 POKE B,C
2780 POKE (B+1),A
2800 NEXT A
2820 RAND USR 16788
2840 LET V=V+1
2860 GOTO 1460
3000 CLS
3020 PRINT AT 3,6;"VOLTA NUMERO
";V-1
3040 IF V-1=0 THEN PRINT AT 3,6;
"LARGADA"
3050 PRINT AT 5,2;"QUAL O NUMERO
 DO ";F$;" ? ";
3060 POKE 16418,2
3080 INPUT C
3100 RAND USR 16737
3110 PRINT C
3120 LET B=VAL X$
3140 PRINT AT 10,4;PEEK (B+2);"■
 LUGAR, ";PEEK B;" VOLTA(S)"
3160 PRINT AT 19,0;"P/ MENU,PRES
SIONE QUALQUER TECLA"
3180 RAND USR 16737
3200 IF INKEY$="" THEN GOTO 3200
3220 GOTO 1460
4000 PRINT AT L,C+3;"■■"
4020 IF N<1 THEN LET N=1
4040 LET L=L-1
4060 IF L<3 THEN LET L=3
4080 SLOW
4100 GOTO 2100
4120 FAST
4140 LET N=N-1
4160 IF C=2 AND N<1 THEN LET N=1
4180 IF C=8 AND N<19 THEN LET N=
19
4200 IF C=14 AND N<19 THEN LET N
=20
4220 IF C=20 AND N<19 THEN LET N
=20
4240 GOTO 4000
5000 FAST
5020 CLS
5040 PRINT AT 0,10;"VOLTA=";V-1
5060 IF V-1=0 THEN PRINT AT 0,10
;"LARGADA "
5080 PRINT "POSICAO ";F$;" VOLTA
(S)"
5100 FOR Y=1 TO F
5120 FOR C=1 TO F
5140 LET B=VAL X$
5160 LET W=PEEK (B+2)
5180 IF W<>Y THEN NEXT C
5200 IF Y=19 OR Y=38 OR Y=57 THE
N GOTO 5260
5220 PRINT TAB 3;Y;"■";TAB 10;C;
TAB 17;PEEK B
5240 NEXT Y
5260 RAND USR 16737
5280 SLOW
5300 PRINT AT 21,3;"QUALQUER TEC
LA CONTINUA";AT 21,3;"QUALQUER T
ECLA CONTINUA"
5320 IF INKEY$="" THEN GOTO 5300
5340 FAST
5360 CLS
5380 PRINT TAB 10;"VOLTA=";V-1
5400 IF V=0 THEN PRINT AT 0,10;"
LARGADA"
5420 PRINT "POSICAO ";F$;" VOLTA
(S)"
5440 IF Y=F+1 THEN GOTO 1460
5460 GOTO 5220
```

```
6000 FAST
6020 RAND USR 16725
6040 LET V=0
6060 GOTO 1460
8000 CLS
8020 PRINT AT 5,5;"LIGUE O GRAVA
DOR";AT 9,5;"PRESSIONE QUALQUER
TECLA"
8040 RAND USR 16737
8060 SLOW
8080 IF INKEY$="" THEN GOTO 8080
8090 SAVE "CONTROLE DE CORRIDA"
9998 POKE 16418,2
9999 RUN
```

PROGRAMA PRINCIPAL

ENDERECO	HEXADECIMAIS				
16725	06	D2	21	82	40
16730	36	00	23	10	FB
16735	C9	00	00	2A	0C
16740	40	23	06	16	0E
16745	21	7E	FE	76	28
16750	03	C6	80	77	23
16755	0D	20	F4	10	F0
16760	C9	2A	0C	40	11
16765	18	79	01	19	03
16770	ED	B0	C9	00	21
16775	18	79	ED	5B	0C
16780	40	01	19	03	ED
16785	B0	C9	00	DD	21
16790	82	40	16	01	FD
16795	21	82	40	0E	00
16800	1E	00	7B	C6	01
16805	FE	2E	28	47	5F
16810	BA	28	02	18	08
16815	FD	23	FD	23	FD
16820	23	18	EB	DD	7E
16825	00	FD	BE	00	28
16830	08	30	EE	79	C6
16835	01	4F	18	E8	DD
16840	7E	01	FD	BE	01
16845	28	12	FE	00	28
16850	DC	FD	7E	01	FE
16855	00	28	E7	DD	BE
16860	01	38	E2	18	CE
16865	DD	7E	02	FD	BE
16870	02	38	C6	79	FE
16875	2D	28	C1	18	D1
16880	79	C6	01	DD	77
16885	02	7A	C6	01	FE
16890	2E	28	09	DD	23
16895	DD	23	DD	23	57
16900	18	94	C9		

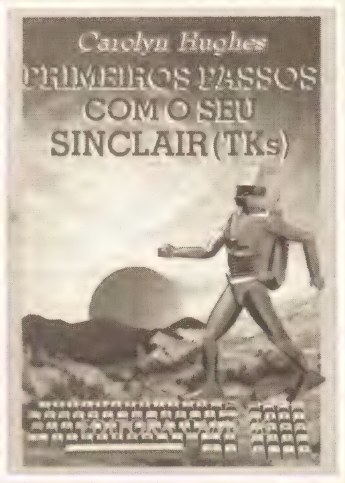
CÓDIGOS HEXADECIMAIS

```
 10 FOR A=16725 TO 16902
 20 INPUT U$
 30 LET U=16*CODE U$+CODE U$(2)
-476
 40 POKE A,U
 45 SCROLL
 50 PRINT A;"=";U$
 60 NEXT A
 70 STOP
```

PROGRAMA CARREGA

MANOEL SILVA RODRIGUES

SE VOCÊ ESTÁ DANDO OS PRIMEIROS PASSOS NA PROGRAMAÇÃO, ESTE LIVRO É PARA VOCÊ



Descubra OS SEGREDOS DO SEU TK

Aprenda A FALAR EM BASIC COM SEU EQUIPAMENTO

Exercite SUAS HABILIDADES COM PROGRAMAS ESPECIAIS

Movimente AS FIGURAS PELA TELA

Divirta-se COM SEIS EMPOLGANTES PROGRAMAS DE JOGOS

JÁ É TEMPO DE SABER TUDO QUE SEU MICRO PODE FAZER. COM SUA AJUDA, NATURALMENTE.

MAIS UM LANÇAMENTO DA EDITORA CAMPUS

PROG

INTEG

CALCULE A ÁREA DE UMA CURVA PELO MÉTODO DOS TRAPÉZIOS

Este programa, "INTEGRAL", calcula a área de uma curva pelo método dos trapézios. O programa pede a função, os limites superior e inferior do intervalo e o incremento, fornecendo a área sob a curva. Do incremento escolhido dependem a precisão do resultado e o tempo de processamento:

Exemplo:

A integral definida da função seno X no intervalo de 0 a π é:

$$\int_0^{\pi} \text{seno } x\, dx = -[\text{cosseno } x]_0^{\pi} = -(\cos \pi - \cos 0) = 2$$

No TK:

Incremento	Resultado	Tempo
0,1	1,9953899	2 seg
0,01	1,9999741	16 seg
0,001	1,9999991	2 min 47 seg

Obs.: O programa aceita funções em x.

Exemplo do uso do programa:

```
LIMITE SUPERIOR ? 10
LIMITE INFERIOR ? 0
INCREMENTO ?      ,01
FUNCAO ?          X**(3/2)
```

MARCEL G. DE AI

RAMA

GRAL

```
INTEGRAL DE X**(3/2)
NO INTERVALO DE 0 A 10
  = 126.17539

OUTRA FUNCAO ? (S/N)
```

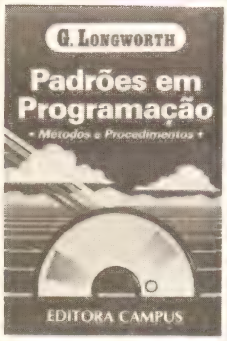
LISTAGEM – **INT-2**

```
   1 REM INTEGRAL - MARCEL
   2 PRINT AT 11,0;"INTEGRACAO -
 METODO TRAPEZOIDAL"
   3 PAUSE 200
   4 CLS
   5 LET FUNC=0
   6 PRINT "LIMITE SUPERIOR ? ";
   7 INPUT B
   8 PRINT B
   9 PRINT ,,"LIMITE INFERIOR ?
";
  10 INPUT A
  11 PRINT A
  12 PRINT ,,"INCREMENTO ?";TAB
 18;
  13 INPUT PR
  14 PRINT PR
  15 PRINT ,,"FUNCAO ?";TAB 13;
  16 INPUT F$
  17 PRINT F$
  18 PAUSE 200
  19 LET X=A
  20 LET FUNC=(VAL F$)/2+FUNC
  21 LET X=B
  22 LET FUNC=(VAL F$)/2+FUNC
  23 FOR X=A+PR TO B-PR STEP PR
  24 LET FUNC=(VAL F$)+FUNC
  25 NEXT X
  26 LET AREA=PR*FUNC
  27 CLS
  28 PRINT "INTEGRAL DE ";F$
  29 PRINT ,,"NO INTERVALO DE ";
A;" A ";B
  30 PRINT ,," = ";AREA
  31 PRINT AT 20,1;"OUTRA FUNCAO
 ? (S/N)"
  32 INPUT A$
  33 IF A$="S" THEN GOTO 4
  34 STOP
  35 SAVE "INTEGRA"
  36 RUN
```

LISTAGEM – **INT-1**

LBUQUERQUE

...MAS, SE VOCÊ JÁ É UM TÉCNICO EXPERIENTE EM BUSCA DE APERFEIÇOAMENTO, ESTE É O LIVRO PARA VOCÊ.



Cobre TODAS AS ETAPAS DO PROCESSO QUE SE INICIA COM O PROJETO E DESENVOLVIMENTO DOS PROGRAMAS, PASSA PELAS DE CODIFICAÇÃO E TESTE E SE ESTENDE ÀS DE IMPLEMENTAÇÃO, MANUTENÇÃO, REVISÃO E OPTIMIZAÇÃO

Apresenta TÉCNICAS ADEQUADAS À MAIORIA DAS INSTALAÇÕES

Contém CHECKLISTS, EXEMPLOS DE DOCUMENTAÇÃO, CONSELHOS E SUGESTÕES

FAÇA DA PROGRAMAÇÃO UMA ATIVIDADE AINDA MAIS SIMPLES E GRATIFICANTE

MAIS UM LANÇAMENTO DA EDITORA CAMPUS

ALÉM DO BASIC

Mais uma coluna sobre Linguagem de Máquina para Linha Sinclair? Não é esta, certamente, a proposta de Nelson do Nascimento Silva dos Santos que, evitando as "fórmulas mágicas", se dispõe a esclarecer dúvidas e desvendar mistérios através de uma abordagem simples e didática, especialmente voltada para o leitor de MICROBITS.

Ilustração: Otavio Studart

Neste nosso segundo encontro, vamos estudar o arquivo de imagem, e como acessá-lo em linguagem de máquina. Isto aumentará muito a nossa flexibilidade.

Podemos encarar a tela como um retângulo de 24 (linhas) x 32 (colunas) = 768 posições. No entanto, os micros de lógica Sinclair arquivam a tela de maneira completamente diferente.

O arquivo de tela é seqüencial, iniciando-se por um carácter 118 (NEW LINE), seguido de 32 bytes para a 1ª linha, um carácter 118 para indicar o fim da 1ª linha, 32 bytes para a 2ª linha, um carácter 118, 32 bytes para a 3ª linha ..., e assim por diante até a 24ª linha (*as duas linhas de edição pertencem ao arquivo de imagem*).

O endereço inicial deste arquivo é *variável*, sendo guardado pela variável do sistema D.FILE (display file = arquivo de imagem), cujos endereços são 16396 e 16397 (400C e 400D). Logo, este endereço pode ser obtido através do comando BASIC

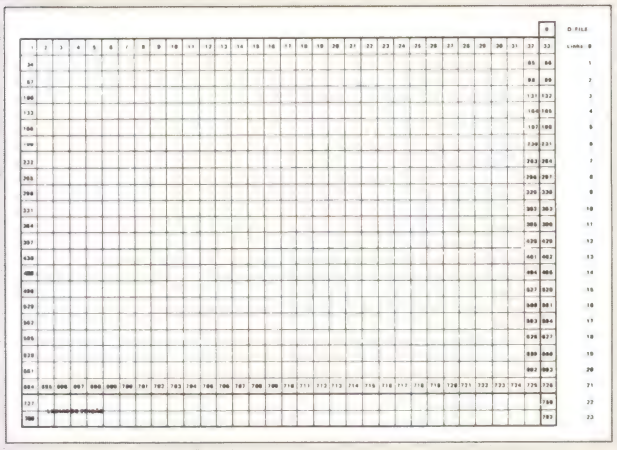
```
PRINT PEEK 16396+256*PEEK 16397
```

Sabemos que o conteúdo deste endereço é 118, um NEW LINE. Assim, se você digitar

```
PRINT PEEK (PEEK 16396+256*PEEK 16397)
```

a resposta será sempre 118.

O esquema da figura abaixo ajudará você a compreender a tela.



O número na quadrícula deve ser *somado* ao endereço indicado por D.FILE para obter-se o endereço da posição correspondente da tela. Observe bem o esquema, e entenda as explicações abaixo.

a) O conteúdo dos endereços relativos às quadrículas hachuradas é sempre 118, e *não pode ser alterado*, ou o micro se perderá, sem saber construir a tela. Em outras palavras: CRASH.

Rode este programinha BASIC:

```
10 LET DF=PEEK 16396+256*PEEK 16397
20 FOR L=0 TO 24
30 PRINT PEEK (DF+33*L),
40 NEXT L
```

Sua saída é: 25 números 118! Ou seja, o NEW LINE inicial do arquivo e mais os 24 NEW LINEs terminais de cada linha.

Um parêntese importante. Se você tem apenas 2K de RAM (TK82 ou 83 *sem* expansão de memória), este programa *NÃO FUNCIONOU*! O motivo é o seguinte: quando o micro tem menos de 3,25K de RAM, ao ser ligado ele não expande a memória da região da tela, deixando-a totalmente comprimida para economizar memória.

Como contornar o problema? Enganando o micro! Vamos alterar o valor de RAMTOP, para fazê-lo "pensar" que tem mais do que 4K. Acrescente (se você está com 2K de RAM apenas) a linha

```
5 POKE 16389,85
```

e rode novamente o programa.

b) É possível imprimir na tela, fazendo POKE nas posições permitidas. Isto é mais rápido do que usar PRINT, porque o sistema operacional do micro não perde tempo verificando a correção das coordenadas. Vamos aos primeiros exemplos e exercícios.

EXEMPLO 1: Colocar um asterisco na primeira posição da tela. O BASIC normal

```
10 PRINT AT 0,0;"*"
```

pode ser substituído, usando D.FILE, por

```
10 LET DF=PEEK 16396+256*PEEK 16397
20 POKE DF+1,23
```

Em linguagem de máquina, teríamos

PROGRAMA 01

0 REM 7 caracteres quaisquer

LD HL,(16396)	2A	0C	40	faz de HL ponteiro para o arquivo da tela
INC HL	23			salta o NEW LINE inicial do arquivo, apontando para a primeira posição "de impressão"
LD (HL),23	36	17		coloca um * nesta posição
RET	C9			volta ao BASIC

Para rodar o programa, RAND USR 16514.

Ex. 01 Qual comando POKE xxxxx,xxx deve ser digitado (modo direto) para que RAND USR 16514 coloque um ⊛ na tela?

EXEMPLO 2: Colocar um asterisco em 10,15.

Em BASIC normal teríamos

```
10 PRINT AT 10,15;"*"
```

Naturalmente não poderemos fazer INC HL, INC HL, INC HL, ..., até alcançarmos a posição 10,15. É necessário calcular o número que deve ser somado a DF. Chamando de L a linha e de C a coluna, temos

33*L+C+1

Em nosso caso,

33*10+15+1=346

Rode o programa

```
10 LET DF=PEEK 16396+256*PEEK 16397
20 PRINT AT 10,15;"■"
30 POKE DF+346,23
```

Após o RUN, só há o asterisco na tela, e a indicação 0/30.

Em linguagem de máquina, seria assim:

PROGRAMA 02

```
0 REM 10 caracteres quaisquer
LD HL,(D.FILE)  2A  0C  40
LD BC,346       01  5A  01
ADD HL,BC       09
LD (HL),23      36  17
RET             C9
```

Rode o seguinte programa BASIC:

```
0  REM aqui está o programa 02
10 PRINT AT 10,15;"■"
20 RAND USR 16514
```

Que ocorreu?

Ex. 02 Fazendo três comandos diretos POKE xxxxx,xxx, altere o programa 02 para que coloque um ▣ no final da última linha de edição, ou seja, em 23,31. Isto poderia ser feito diretamente, através de PRINT AT 23,31;" ▣ "? Como este problema pode ser contornado *em BASIC*?

EXEMPLO 3: Fazer uma diagonal na tela, desde 0,0 até 21,21. O carácter de construção será Graphics shift Y.

Em BASIC normal, seria:

```
10 FOR F=0 TO 21
20 PRINT TAB F;"▚"
30 NEXT F
```

Usando o arquivo de tela, a solução mais simples seria

```
10 LET DF=PEEK 16396+256*PEEK 16397
20 FOR F=0 TO 21
30 LET S=34*F
40 POKE DF+1+S,134
50 NEXT F
```

Observe atentamente o esquema da tela, e certifique-se de ter entendido o *34* da linha 30.

Convertendo para linguagem de máquina, teremos o

PROGRAMA 03

```
0 REM 15 caracteres quaisquer
      LD HL,(D.FILE) 2A  0C  40
      INC HL         23
      LD DE,34       11  22  00
      LD B,22        06  16        B = contador
                                   para 22 linhas
POKE  LD (HL),134    36  86
      ADD HL,DE      19
      DJNZ POKE      10  FB        volta ao rótulo
                                   POKE,
                                   fazendo o loop
      RET            C9
```

Percebeu como o rótulo POKE auxilia a compreender o programa?

Ex. 03

a) Digite, modo direto, POKE 16524,23. Rode (RAND USR 16514) e ... que ocorre? Por quê?
b) Digite POKE 16519,33 e rode. Que ocorre? Por quê?
c) Digite POKE 16519,1 e rode. Que ocorre? Por quê?
d) Digite POKE 16519,34 e depois POKE 16522,24. Rode. Que ocorre? Por quê?
e) Que *ocorreria* se você *digitasse* POKE 16519,32 e *rodasse* o programa? Por quê?

Não prossiga sem ter compreendido *todas* as ocorrências, especialmente as das letras *d* e *e*. É importante não saltar etapas se você quer realmente aprender.

Vamos agora ao programa mais importante desta nossa segunda "aula": encher a tela. Observe atentamente a estrutura do programa — *é a mais curta possível, e será usada em inúmeros programas.*

PROGRAMA 04

```
0 REM 17 caracteres quaisquer
      LD HL,(D.FILE) 2A  0C  40

      LD B,22        06  16        B = contador
                                   para 22 linhas
LINHA INC HL         23            salta NEW
                                   LINE
      LD C,32        0E  20        C = contador
                                   para 32 colunas
```

```
POKE    LD (HL),136     36  88
        INC HL          23          próxima posição
        DEC C           ØD
        JR NZ,POKE      2Ø  FA      completa a
                                    linha
        DJNZ LINHA      1Ø  F5      enche a tela
        RET             C9
```

RAND USR 16514 enche a tela de cinza com incrível velocidade.

Ex. Ø4

Naturalmente o programa Ø4 pode ser escrito com os contadores invertidos: C para linhas e B para colunas. Faça isto — também leva 17 bytes.

Ex. Ø5

Acrescentando ao seu programa acima CALL PRINT AT Ø,Ø (CALL Ø8F5, lembra da coluna passada?) no lugar certo e imprimindo espaços você terá um CLS *muito mais rápido* que o da máquina, com apenas 2Ø bytes. Teste-o com o seguinte programa BASIC.

```
 Ø REM aqui está o seu programa
1Ø FOR F=Ø TO 1Ø
2Ø PRINT TAB F*2.3;"MICROBITS"
3Ø PRINT TAB F*2.3;" ▬▬▬▬ "
4Ø NEXT F
5Ø CLS
6Ø PRINT "TELA LIMPA"
```

Observe o tempo levado pelo CLS. Agora substitua a linha 5Ø por RAND USR 16514 e aprecie a incrível diferença de velocidade!

Até a próxima!

RESPOSTAS DOS EXERCÍCIOS DO Nº ANTERIOR

Ex. Ø1

O programa ocupa 22 bytes.

```
        16514 Ø1  ØØ  ØØ   LD BC,Ø
        16517 CD  F5  Ø8   CALL 2293 (PRINT AT)
        16520 3E  ØØ       LD A,Ø
        16522 Ø6  BØ       LD B,176
PRINT   16524 D7           RST 1Øh
        16525 D7           RST 1Øh
        16526 D7           RST 1Øh
        16527 D7           RST 1Øh
        16528 1Ø  FA       DJNZ PRINT
        16530 ØE  ØØ       LD C,Ø
        16532 CD  F5  Ø8   CALL 2293 (PRINT AT)
        16535 C9           RET
```

Quanto à velocidade, *não* há ganho.

Ex. Ø2

O programa ocupa 9 bytes.

```
        16514 3E  1C       LD A,28
PRINT   16516 D7           RST 1Øh
        16517 3C           INC A
        16518 FE  2C       CP 44
        16520 2Ø  FA       JR NZ,PRINT
        16522 C9           RET
```

Ex. Ø3

O programa ocupa 9 bytes.

```
        16514 3E  3F       LD A,63
PRINT   16516 D7           RST 1Øh
        16517 3D           DEC A
        16518 FE  25       CP 37
        16520 2Ø  FA       JR NZ,PRINT
        16522 C9           RET
```

Ex. Ø4

O programa ocupa 15 bytes.

```
        16514 Ø1  Ø3  ØB   LD BC,... (B=11,C=3)
        16517 CD  F5  Ø8   CALL PRINT AT
        16520 3E  BF       LD A,191
PRINT   16522 D7           RST 1Øh
        16523 3D           DEC A
        16524 FE  A5       CP 165
        16526 2Ø  FA       JR NZ,PRINT
        16528 C9           RET
```

Ex. Ø5

O programa ocupa 22 bytes.

```
        16514 Ø1  ØØ  Ø9   LD BC,... (B=9,C=Ø)
        16517 CD  F5  Ø8   CALL PRINT AT
        16520 3E  ØØ       LD A,Ø
PRINT1  16522 D7           RST 1Øh
        16523 3C           INC A
        16524 FE  4Ø       CP 64
        16526 2Ø  FA       JR NZ,PRINT1
        16528 87           ADD A,A
PRINT2  16529 D7           RST 1Øh
        16530 3C           INC A
        16531 FE  CØ       CP 192
        16533 2Ø  FA       JR NZ,PRINT2
        16535 C9           RET
```

Não consigo escapar do espírito da nova seção DESAFIO... Lá vai: você saberia reduzir este programa para *apenas* 2Ø bytes, mantendo o *mesmo* efeito? Sugestão: use INC A e RST 1Øh uma única vez cada um. PENSE! Solução no próximo nº, quando publicaremos os nomes dos acertadores.

TRON

Participe de uma corrida no interior do seu micro.
Mas fique de olho no TRON e muita atenção, pois qualquer distração poderá ser fatal!

UMA FANTÁSTICA AVENTURA

Este jogo para micros compatíveis com Sinclair com apenas 2 Kbytes de memória, foi baseado nas cenas do filme "TRON" (um jovem programador foi "engolido" pelo computador e passou a viver entre seus circuitos, onde ficou conhecido como "TRON" e vive aventuras incríveis, como por exemplo a disputa na qual se baseia este programa).

Inicialmente é delimitada a área onde V. disputará uma corrida TRON (que vive dentro do seu micro). Após dada a largada, V. deve evitar esbarrar nos limites da área ou ultrapassar um caminho já percorrido, que vai sendo marcado na tela.

Cada um possui 10 chances para errar.

Preparação

Digite a listagem 1 e então a listagem 2. Observe que a listagem só possui comandos sem numeração no início da linha.

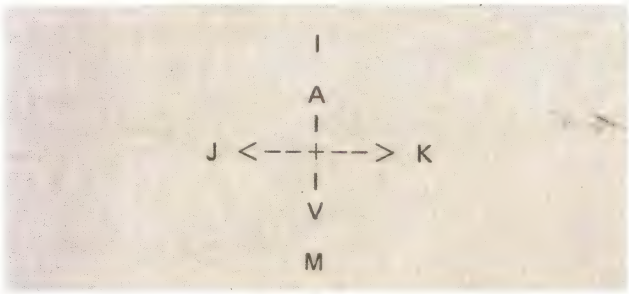
Prepare seu gravador para gravar o programa (conecte os cabos, ajuste o volume etc.) e digite < GOTO 500 > ao mesmo tempo que V. ativa o gravador.

O Jogo

Quando a gravação terminar aparecerá a mensagem "seu nome?". Digite seu nome (máximo de 10 letras) e em poucos segundos aparecerá a área do "duelo" na tela do micro. V. é o "O" em fundo inverso e TRON o "*" em fundo inverso.

Se ocorrer algum erro, corrija-o e use < GOTO 10 > (nunca use < RUN >!) e grave-o novamente como descrito. Depois, para executá-lo novamente, basta dar < LOAD "TRON" > e o programa logo inicia solicitando seu nome (não é necessário dar < RUN >!).

O Quadro 1 mostra as teclas que devem ser usadas para mudar sua direção.



Quadro 1

Por exemplo, se V. estiver indo da esquerda para a direita (sentido inicial) e quiser subir, aperte a letra "I" até observar que V. mudou de direção (cuidado para não apertar uma tecla diferente de "I", "J", "K" ou "M"!).

Evite esbarrar nos cantos ou nos caminhos que vão sendo marcados (quadrado em fundo inverso).

No alto da tela do micro sempre aparecem os pontos de cada um. Cada vez que um esbarra num quadrado já ocupado, este quadrado pisca indicando o local do "choque" e é marcado um ponto a mais para o adversário. Ganha quem primeiro fizer 10 pontos.

No final do jogo, o micro pára esperando que uma tecla qualquer seja pressionada para dar início a um novo jogo, solicitando o nome do adversário do "TRON".

Macetes

Como os pontos são contados quando se esbarra no caminho, V. pode então atravessar o caminho do "TRON" ou até mesmo andar junto dele. Isto é útil quando V. estiver sendo encurralado pelo "TRON".

Procure encurralar o "TRON", não deixando alternativas para a sua fuga nos labirintos que vão se formando.

Finalmente, 2 conselhos: Fique de olho no "TRON" que pode mudar de direção repentinamente e muita atenção, pois qualquer distração poderá ser fatal!

```
 10  LET E=0
 20  LET F=0
 30  CLS
 40  PRINT "SEU NOME?"
 45  INPUT N$
 50  FAST
 60  CLS
 70  FOR I=0 TO 31
 80  PRINT AT 1,I;CHR$ x;AT 21,I
;CHR$ X
 90  IF I<22 THEN PRINT AT I,0;C
HR$ X;AT I,31;CHR$ X
100  NEXT I
200  LET M=PEEK 16396+256*PEEK 1
6397
210  LET A=M+335
220  LET B=M+356
230  LET C=3
240  LET D=2
250  PRINT AT 0,8;"TRON ";F;" -
";E;"  ";N$
290  SLOW
300  LET A$=INKEY$
310  IF A$<>"" THEN LET C=(A$>="
I") + (A$>="J") + (A$>="K") + (A$>="M"
)
320  POKE A,X
330  LET A=A+A(C)
340  IF PEEK A=K THEN GOTO 630
350  POKE A,180
400  IF RND>.1 AND PEEK (B+A(D))
<>X THEN GOTO 460
410  LET R=INT (RND*17)
420  FOR I=R TO R+3
425  LET D=I-INT (I/4)*4+1
430  IF PEEK (B+A(D))<>X THEN GOTO 460
440  NEXT I
450  GOTO 600
460  POKE B,X
470  LET B=B+A(D)
480  POKE B,151
490  GOTO 300
500  SAVE "TRON"
510  GOTO 10
600  LET A=B
610  LET E=E+1
620  LET F=F-1
630  LET F=F+1
640  FOT I=1 TO 50
650  POKE A,0
660  POKE A,8
670  NEXT I
680  IF E<>10 AND F<>10 THEN GOTO 50
690  IF INKEY$="" THEN GOTO 690
700  GOTO 10
```

```
DIM A(4)
LET A(1)=-33
LET A(2)=-1
LET A(3)=1
LET A(4)=33
LET X=128
```

Bacharel em Informática pela UFRJ, cursando mestrado na COPPE/UFRJ, trabalha como Analista de Sistemas na Portobrás.

MAURICIO COSTA REIS

APROXIMAÇÃO DE ROMBERG

Este programa de integração numérica utilizando a aproximação de Romberg foi baseado no livro PROCESSAMENTO DE DADOS, por J.A.R. Santos e W.M. Stravinsky — publicado por LTC e EDUSP.

Uma função pode ser integrada por esse método.

$$\int_a^b f(x)\,dx \simeq \begin{cases} I_R^0(h) \\ I_R^0(h/2) & I_R^1(h) \\ I_R^0(h/4) & I_R^1(h/2) & I_R^2(h) \\ \vdots & \vdots & \vdots \end{cases}$$

Figura 1

Onde $I_R^0(h) = I_T(h)$ é a integral calculada pelo método dos trapézios (e onde $h = (b-a)/n$; n sendo o número de intervalos) e

$$I_R^1(h) = \frac{4I_T(h/2) - I_T(h)}{3} - \frac{3}{4} C_4 h^4 + \cdots$$

onde a integral de Romberg, de ordem 1, tem erro proporcional a h^4; a de ordem 2 tem erro proporcional a h^6.

A fórmula geral é:

$$I_R^n(h/2^k) = \frac{4^n I_R^{n-1}(h/2^{k+1}) - I_R^{n-1}(h/2^k)}{4^n - 1}$$

E para um cálculo correto deve-se efetuar os cálculos fila por fila como na figura 1.

As variáveis usadas no programa são:

F$	—	armazena a função f(x) em forma de "string"
XØ	—	limite inferior da integral
XM	—	limite superior da integral
E	—	número de intervalos
N	—	ordem de integração
R(N, N)	—	matriz onde se coloca os resultados intermediários para os cálculos

Desempenho (em FAST)

função usada: $\int_0^1 \sqrt{x}\,dx = 0{,}6666666\overline{6}$ (valor exato)

E = 2	valor	tempo	E = 10	valor	tempo
N = 1	0,6565269	3 seg	N = 1	0,665657	3 seg
N = 2	0,6635178	4 seg	N = 2	0,66638	11 seg
N = 3	0,665685	7 seg	N = 3	0,666569	18 seg
N = 4	0,666287	10 seg	N = 4	0,6666327	45 seg
N = 5	0,66653	22 seg	N = 5	0,6666547	1:30

OBS: Pode-se economizar memória se em vez de dimensionar uma matriz (triangular inferior), colocar em uma forma determinada, num vetor. Também o tempo de cálculo pode ser reduzido se testar cada cálculo obtido com o anterior e parar se estiver dentro da precisão requerida.

Por outro lado, esse programa é pequeno, rápido e de grande precisão, podendo ser usado como sub-rotina de outros programas. A listagem a seguir mostra um programa desprovido de qualquer auto-explicação.

```
  0 REM  EDUARDO MARCHIORI
  1 REM  INTEGRACAO NUMERICA
  2 REM      DE  ROMBERG
  4 PRINT "ENTRE COM A FUNCAO C
OMO Y=F(X)"
  5 INPUT F$
  6 PRINT "
                              "
  7 PRINT AT 4,0;"Y = ";F$
  8 PRINT AT 6,0;"
                              "
  9 PRINT "ENTRE COM O LIM. INF
ERIOR"
 10 INPUT X0
 12 PRINT "X0 = ";X0
 14 PRINT "ENTRE COM LIM. SUPER
IOR"
 15 INPUT XM
 17 PRINT "XM = ";XM
 19 PRINT "ENTRE COM O Nª DE IN
TERVALOS"
 20 INPUT E
 22 PRINT "E = ";E
 24 PRINT "ENTRE COM ORDEM INTE
GRACAO"
 25 INPUT N
 26 PRINT "N = ";N
 30 LET N=N+1
 35 DIM R(N,N)
 40 CLS
 44 FAST
 45 FOR I=1 TO N
 50 GOSUB 200
 55 LET R(I,1)=SOMA
 60 LET E=E*2
 65 NEXT I
 70 LET M=1
 75 FOR J=2 TO N
 80 FOR I=M TO N-1
 85 LET R(I+1,J)=(4**(J-1)*R(I+
1,J-1)-R(I,J-1))/(4**(J-1)-1)
 90 NEXT I
 95 LET M=M+1
100 NEXT J
105 PRINT AT 10,0;"O VALOR DA I
NTEGRAL E ",AT 12,5;R(N,N)
110 STOP
190 REM SUB INTEGR. TRAPEZOIDAL
200 LET H=(XM-X0)/E
205 LET SOMA=0
210 FOR J=1 TO E-1
215 LET X=X0+J*H
220 LET SOMA=SOMA+VAL F$
225 NEXT J
230 LET X=X0
235 LET Y1=VAL F$
240 LET X=XM
245 LET Y2=VAL F$
250 LET SOMA=H*(SOMA+(Y1+Y2)/2)
255 RETURN
```

EDUARDO MARCHIORI

Notícias

• A IBM americana acaba de produzir uma máquina que "reconhece" sentenças faladas em inglês. Segundo a IBM, o aparelho dispõe de um dicionário de 5000 palavras, e, embora necessite que o locutor deixe um breve espaço entre cada palavra, reconhece aproximadamente 95% delas. Distingue até mesmo homônimos (palavras que têm o mesmo som) como 'to', 'too' e 'two', dependendo do contexto em que estão inseridos.

• A firma inglesa ICL (International Computers Limited) lançou recentemente o seu micro OPD ("One Per Desk" ou "Um em cada mesa"). É um desenvolvimento baseado no Sinclair QL, com especificações bastante impressionantes. Para quem nunca ouviu falar do QL, este micro tem as seguintes especificações:

- 128K bytes de ROM (expandível até 640K)
- Processador de 32 bits (MOTOROLA 68008)
- Sistema operacional QDOS com recursos de multiprogramação
- Dois "microdrives" (unidade de fita contínua que funciona como diskette) que possibilitam o armazenamento de até 100K bytes cada
- Uma nova linguagem SUPERBASIC com mais recursos que o BASIC do ZX81 e ZX SPECTRUM
- ROM de 32K
- Quatro softwares profissionais de PSION incluídos: Processador de texto, planilha eletrônica, gerenciador de banco de dados e sistema gráfico.

E o que tem o OPD a mais? O que realmente destaca este micro de qualquer outro é o sistema embutido de comunicação vocal e de dados. As especificações resumidas do OPD são:

- 128K de ROM de software de sistema em 4 chips ICL de 32K bytes cada
- Um chip TI TMS 5220 de sintetizador de voz
- Um ROM com vocabulário para o sintetizador
- Dois diretórios telefônicos – 1 para chamadas de serviços computadorizados, e outro para chamadas normais (voz a voz)
- Telefone embutido na própria unidade de teclado
- 128K de RAM dinâmica
- 2K de RAM estática com parâmetros essenciais do sistema. Os dados são mantidos através de uma pilha de lítio com duração de 5 anos
- 208K de ROM em cartucho com os quatro softwares de PSION mencionados acima para o QL.
- 2 microdrives, de 95K cada
- Modem
- Calculadora simples de 16 dígitos de precisão

O micro em si não possui chave liga/desliga. Foi projetado para ser ligado continuamente para atender chamadas telefônicas. O único interruptor é do vídeo, que pode ser desligado para aumentar a sua vida útil. Mesmo assim, se for deixado ligado por mais que 5 minutos, sem qualquer atividade, a tela é apagada para proteger a camada de fósforo.

Há um alto-falante embutido que permite monitorizar chamadas sem levantar a unidade do telefone. Mas para falar é necessário usar o telefone mesmo – o OPD não possui microfone.

Vários programas podem ser executados simultaneamente. Assim, o usuário não precisa esperar a execução de um aplicativo demorado para utilizar.

O OPD atende o telefone, recebendo chamadas vocais ou de dados digitais, sem interferir nas aplicações em execução. Atende o telefone até quando o usuário está fora do escritório, e pode ser programado para responder com uma mensagem vocal através do sintetizador de voz.

Para armazenar esta mensagem, é utilizado um vocabulário de 200 palavras. A mensagem é criada simplesmente digitando o texto no teclado. Qualquer palavra não reconhecida é destacada na tela, para ser corrigida em seguida.

E o preço? O OPD básico, com monitor de nove polegadas, é de £1150 (por volta de US$1200), mais £150 para o cartucho de software PSION. Por este preço não há nenhum outro micro que possa competir com ele. A ICL espera vender 250.000 unidades em 1985 – será isto um sonho? Veremos mais para o final do ano como andam as vendas.

MICROBITS

UMA BOA NOTÍCIA PARA VOCÊ:
Os leitores que assinam MICROBITS economizam tempo e dinheiro!

1) LEMBRE-SE: Você irá receber cada número comodamente em sua casa, tão logo seja publicado.
2) COMPARE: Preço Assinatura (6 números) ..... Cr$ 27.000
Preço Nº Avulso ..... Cr$ 6.500
3) SOLICITE: Ainda hoje a sua assinatura.

MICROBITS
PEDIDOS DE ASSINATURA PARA:
Editora Campus Ltda.
Rua Barão de Itapagipe, 55 - Rio Comprido - 20261 Rio de Janeiro - RJ - Tel.: (021) 284 8443

PROGRAMA

O programa a seguir foi projetado para o usuário radioamador criar uma lista de prefixos e nomes, os quais são colocados em ordem alfabética de acordo com o prefixo.

Com 16K de RAM é possível armazenar aproximadamente 200 prefixos.

Facilmente o leitor poderá modificar o programa para os mais variados tipos de aplicação.

Quando rodado o programa, aparece o "menu" dos comandos (Figura 1).

Um exemplo de entrada de dados é dado na Figura 2, e a listagem dos prefixos e nomes já ordenados na Figura 3.

ORDENADOR

ATENÇÃO RADIOAMADOR:

CRIE SUA LISTAGEM ALFABÉTICA DE PREFIXOS E NOMES

```
   1 REM "PROGRAMA DESENVOLVIDO
POR RENATO STRAUSS E SINESIO AMO
RIM FLHO PARA RADIOAMADORES"
   2 REM "RENSI ELETRONICA DIGIT
AL CAIXA POSTAL 8551 CEP 01051 S
AO PAULO SP"
   5 PRINT TAB 12;"
   6 PRINT TAB 12;"  RENSI  "
   7 PRINT TAB 12;"
  10 PRINT ,,,,,,TAB 11;"*******
****"
  11 PRINT TAB 11;"*ORDENADOR*"
  12 PRINT TAB 11;"***********"
  20 LET B=1
  25 LET X=200
  27 DIM A$(2,4)
  30 DIM C$(1,2,7)
  35 DIM X$(X,15
  45 PRINT ,,,"QUANTOS PREFIXOS
 DESEJA ORDENAR?
  46 PRINT TAB 9;"P/ COMANDOS R
O?"
  50 INPUT N
  51 IF N=0 THEN PRINT AT 11,0;"
"
  52 PRINT TAB 9;"
"
  55 IF N=0 THEN GOTO 343
  58 LET N=N+B-1
  60 IF N>X THEN PRINT ,,"CAP. A
TE 200..."
  70 IF N>X THEN GOTO 50
  90 CLS
 100 PRINT ,,"PREFIXO  "
 120 INPUT C$(1,1)
 121 PRINT C$(1,1)
 125 PRINT "NOME (ORA)  ";
 130 INPUT C$(1,2)
 131 PRINT C$(1,2)
```

```
 140 LET X$(B)=C$(1,1)+" "+C$(1,
2)
 145 PRINT ,,
 150 PRINT B;TAB 4;X$(B)
 160 IF PEEK 16442=2 THEN SCROLL

 165 LET B=B+1
 170 IF B<=N THEN GOTO 100
 175 CLS
 180 PRINT "LISTANDO..."
 190 PAUSE 150
 205 FOR K=1 TO N-1
 210 FOR J=1 TO N-K
 215 IF X$(J)<=X$(J+1) THEN GOTO
 250
 220 LET T$=X$(J)
 230 LET X$(J)=X$(J+1)
 240 LET X$(J+1)=T$
 250 NEXT J
 260 NEXT K
 310 CLS
 315 PRINT "              LISTAGEM"
 316 PRINT
 317 IF N>0 THEN GOTO 325
 318 PRINT AT 20,5;"NENHUM REGIS
TRO"
 319 GOTO 341
 325 FOR I=1 TO N
 330 PRINT I;TAB 4;X$(I)
 335 NEXT I
 340 PRINT ,,
 341 PRINT "*PARA CONTINUAR NEW
LINE*"
 342 GOSUB 1000
 343 PRINT ,,"COMANDOS:",,"*PARA
 GRAVAR    DIGITE <S>*"
 344 PRINT "*P/GRAVAR DADOS DIGI
TE <V>*","*PARA CONTINUAR DIGITE
 <C>*"
 345 PRINT "*PARA ALTERAR   DIGI
TE <A>*"
 346 PRINT "*PARA RELISTAR  DIGI
TE <L>*"
 348 GOSUB 1000
 349 IF INKEY$="S" THEN GOTO 950
0
 350 IF INKEY$="C" THEN GOTO 40
 351 IF INKEY$="A" THEN GOTO 520
 352 IF INKEY$="L" THEN GOTO 300
 353 IF INKEY$="V" THEN GOTO 900
0
 354 GOTO 343
 520 PRINT ,,,"    ENTRE COM NUM
ERO DO ITEM              A SER AL
TERADO          "
 525 INPUT Y
 530 PRINT ,,Y;TAB 4;X$(Y)
 540 PRINT ,,"ENTRE NOVO PREFIXO
 $ ";
 550 INPUT C$(1,1)
 551 PRINT C$(1,1)
 555 PRINT "ENTRE NOVO  NOME  $
";
 560 INPUT C$(1,2)
 561 PRINT C$(1,2)
 565 IF X$(Y, TO 7)=C$(1,1) THEN
 GOTO 590
 570 LET X$(Y)=C$(1,1)+" "+C$(1,
2)
 580 GOTO 175
 590 LET X$(Y)=C$(1,1)+" "+C$(1,
2)
 595 GOTO 300
 605 CLS
 610 PRINT "ENTER TODAYS DATE."
 615 INPUT I$
 620 LET P$=I$
 625 LET P=USR 3201
 650 FOR I=1 TO INT (N/2+.5)
 655 LET A$(1)=STR$ I
 660 LET A$(2)=STR$ INT (N/2+I+.
5)
 665 LET P$=A$(1)+X$(I)+" "+A$(2
)+X$(N/2+I)
 670 LET P=USR 8192
 675 NEXT I
 680 GOTO 300
1000 LET J$=INKEY$
1010 IF INKEY$="" THEN GOTO 1010
1020 CLS
1030 RETURN
9000 SAVE "ORDENADOR
9100 GOTO 343
9500 SAVE "ORDENADOR
9990 RUN
```

```
COMANDOS:
*PARA GRAVAR      DIGITE  S *
*P/GRAVAR DADOS   DIGITE  V *
*PARA CONTINUAR   DIGITE  C *
*PARA ALTERAR     DIGITE  A *
*PARA RELISTAR    DIGITE  L *
```

FIGURA 1 – MENU DOS COMANDOS

```
PREFIXO          PY2-EMI
NOME (ORA)       RENATO

1    PY2-EMI RENATO

PREFIXO          PY2-AIM
NOME (ORA)       SINESIO

2    PY2-AIM SINESIO

PREFIXO          PY2-AQO
NOME (ORA)       LUIZ

3    PY2-AQO LUIZ
```

FIGURA 2 – ENTRADA DE DADOS

```
              LISTAGEM

1    PY2-AIM SINESIO
2    PY2-AQO LUIZ
3    PY2-EMI RENATO
4    PY2-NOG GUSTAVO

*PARA CONTINUAR NEW LINE*
```

FIGURA 3 – LISTA COMPLETA

RENATO STRAUSS
SINÉSIO AMORIM FILHO

Classificados

COMPRA, VENDA, TROCA, OFERTAS,
PRESTAÇÃO DE SERVIÇOS.
Classifique como quiser, mas use este espaço, que é seu.

SOFTWARE

1

• **VENDO** ou troco programas de Lógica SINCLAIR. Envie sua lista de programas para: Sérgio S. Caixa, Caixa Postal 79406, CEP 25500 São João de Meriti – RJ. Possuo mais de 500 programas.

• **TROCO** ou vendo programas para SINCLAIR e TRS-80. Mande listagem e envelope selado para resposta. Também quero trocar idéias, informações, dicas, curiosidades e correspondência. Marcelo D. Penalva, Av. 64 nº 2493, CEP 13500 – Rio Claro – SP.

• **TROCO** software, informações, projetos de hardware e ADD-ON's para o SINCLAIR ZX81 e compatíveis. Compro e troco livros e revistas sobre o assunto em pauta. Necessito entrar em contato com quem possua as revistas HOBBY ELECTRONICS dos meses 6, 7, 8 e 12 de 1983. Valério F. Laube, Caixa Postal 30, CEP 89260 – Schroeder – SC.

• **CRED-SOFT** e SÉRIES SOFT-BOX para TK e similares. Novo endereço: SOFT-BOX, Caixa Postal 79406, CEP 25500 – São João de Meriti – RJ.

• **VENDEMOS** e trocamos programas da linha SINCLAIR. Escrevam para: Rua Curuzu 62, Alto da Lapa, CEP 05084 – São Paulo – SP e forneceremos relação dos programas e seus preços.

• **ALTA RESOLUÇÃO** – programas para micros adaptados: Pinball, G. Cósmica, Pac-Man etc. INÉDITOS: Q-Bert, Bat Cage, River Raid, Rev. Colmeia e mais 200 programas disponíveis para troca ou venda. Tratar: Alexandre M. de Brito, SHIN QI-13, Cj. 14/Cs. 4, CEP 71500 – Brasília – DF – Tel. (061) 577-3804.

• **VENDO** e troco soft de comunicação e jogos para APPLE. Luiz Roberto – Tel. (011) 287-4625.

• **VENDE-SE** programas para TK, CP200 e similares. Possuo programas nacionais e importados. Faço descontos especiais. Alexandre Basílio Caputo. Rua Ipurinás, 46 – Brooklin Novo – CEP 04561 – São Paulo – SP – Tel. (011) 240-0636.

• **TROCO** programas para linha SINCLAIR – Tel. (011) 531-3600 e 520-2863. Av. Jônia 213, CEP 04634 – São Paulo – SP, com Rinaldo.

HARDWARE

2

• **NE-Z8000** – Esquema completo com SLOW e expansão de memória (1,2 ORTN); Lay-out da placa SLOW (0,3 ORTN); Listagem da EPROM (0,3 ORTN). Cheque nominal para Jan Martin Lund, Rua Frederico Ozanan 16 bloco A apt? 21, CEP 11100 – Santos – SP.

• **VENDO** projeto de uma impressionante máquina eletrônica, a qual pode ser programada para ensinar línguas, ciências etc. É totalmente transistorizada. Funciona com cartões perfurados. Tratar com Valdeci – Rua Terenzio Galesi 299, Algodoal, CEP 13400 – Piracicaba – SP.

• **VENDO** CP200 com mais de 50 programas (em Assembler e BASIC). Entre em contato com André Luiz S. Brisighello pelo telefone (011) 521-6345.

• **INSTALO** em seu micro da linha SINCLAIR alta resolução, Inversão de vídeo e filtro para HIGH-SPEED. Luiz Roberto – Tel. (011) 287-4625.

• **COMPRO** um TK82 ou TK83. Entre em contato com Wellington Martins Mourão, SQ-15, Q-13 Casa 13, Cidade Ocidental – CEP 77220 – Luziânia – GO.

• **CW NO ZX81 – VENDO** interface (mais fita e manual). Temos 600 programas para escolha. Informações com A. K. Produções – Caixa Postal 77 – CEP 95590 – Tramandaí – RS.

• **RADIOAMADOR, VENDO** interfaces para RTTY e CW para micros compatíveis com o SINCLAIR, TK82C, TK83/85, CP200, ZX81, TIMEX1000. Informações com PY2-EMI Renato Strauss. Rua Cardoso de Almeida 654/32 – CEP 05013 – São Paulo – SP.

• **VENDO** TK85 com 48K, alta resolução, saída para monitor de vídeo, 2 gravadores, joystick e 170 programas. Tratar com Eugênio Betanho. Tel. 64-7074 – São Paulo – SP.

CLUBES

3

• **ESTAMOS** formando um clube para quem possui micros de lógica SINCLAIR e também vídeo-games. Trocaremos revistas, programas etc. Daremos dicas de jogos e muito mais. Marcelo Cláudio Faria, Cond. Raul Renato, Bloco L apt? 14, Jardim Garcia – CEP 13100 – Campinhas – SP.

Dicas

QUANTOS BYTES AINDA RESTAM?

Na digitação de programas longos, esta pergunta se torna freqüente, e às vezes angustiante.

A memória ainda não ocupada pelo seu programa BASIC está entre os endereços apontados por E.LINE e RAMTOP. Se você não sabe destas coisas, recomenda-se a leitura atenta do manual nos tópicos VARIÁVEIS DO SISTEMA e ORGANIZAÇÃO DA ARMAZENAGEM. Observe o esquema abaixo:

ENDEREÇO		Nº de Bytes	Variável do Sistema	Função
Decimal	Hexa			
16388	4004	2	RAMTOP	Endereço do topo da RAM existente ou alterada pelo usuário
16404	4014	2	E.LINE	Ponteiro para a área de trabalho

Assim, em BASIC, para saber o número de bytes "livres", digite o comando direto

```
PRINT (PEEK 16388+256*PEEK 16389)-(PEEK
16404+256*PEEK 16405)
```

Ou então você pode transformar isto num programa em linguagem de máquina, e poupar toda esta digitação trabalhosa. Vejamos como:

```
0 REM 14 caracteres quaisquer
16514  LD HL, (E.LINE)    2A  14  40
16517  PUSH HL            E5
16518  POP BC             C1
16519  LD HL, (RAMTOP)    2A  14  40
16522  AND A              A7
16523  SBC HL, BC         ED 42
16525  PUSH HL            E5
16526  POP BC             C1
16527  RET                C9
```

E agora, para saber o número de bytes restantes na memória de seu micro é só digitar PRINT USR 16514.

Se você for fazer muitas vezes esta consulta, transforme-a em linha de programa, assim:

```
9998  STOP
9999  PRINT USR 16514;"_BYTES_RESTANTES"
```

Acesse esta pequena rotina através de GOTO 9999. Simples, não?

NELSON N. S. SANTOS

COMO ACELERAR A LISTAGEM

Todo usuário SINCLAIR sabe que quando a tela fica cheia, durante a digitação de um programa longo, ocorre uma listagem automática a fim de abrir espaço a cada linha digitada. Para evitar essa perda de tempo, vários métodos são empregados pelos usuários. Pode-se, por exemplo, "Pokear" os endereços 16419 e 16420, colocando neles o nº da última linha digitada, a fim de movê-la para o topo da tela. Porém este método exige que o usuário memorize ou procure no manual estes endereços. Por isto, alguns desenvolveram rotinas em linguagem de máquina que realizam este trabalho mas, neste caso, a rotina terá que ser introduzida antes de iniciar a digitação do programa, sem contar o inevitável consumo de memória. Outros, ainda, digitam em modo FAST, enfrentando o piscar da tela.

Para evitar todas essas desvantagens, utilizo um método muito simples, que acredito ser inédito, pois nunca o vi publicado em revistas especializadas. Portanto, quando a listagem começar a rolar pela tela, entre diretamente com os seguintes comandos:

```
LIST
EDIT
NEW LINE
```

E continue a digitar o programa. Assim a tela ficará limpa, com a última linha digitada antes deste procedimento aparecendo no topo.

GELSON DIAS SANTOS
Porto Alegre – RS

ENTRE UM PROGRAMA E OUTRO SINTONIZE O PROGRAMA

SÁBADO ESPECIAL DE JORGE MASCARENHAS
ENTREVISTAS – MÚSICA JOVEM – BRINDES
RÁDIO BANDEIRANTES/1360 AM
DAS 10 AO MEIO - DIA

Livros

DAVID THOMAS ANDERSON

• **LOIOLA, Caetano R. A.**
Rotinas Matemáticas em BASIC para MICROS
Série SOFTWARE, Editora Campus, 1985.

A série SOFTWARE da CAMPUS é coordenada por Jorge da Cunha Pereira Filho, autor de BASIC BÁSICO, BASIC PARA MICROS PESSOAIS, RPG-II, INTRODUÇÃO À PROGRAMAÇÃO FORTRAN e outras obras.

No livro em questão, o leitor é recomendado a ler os apêndices antes de iniciar a leitura do conteúdo em si. Destes apêndices (A a G), os de A a D são escritos pelo próprio coordenador e demonstram a sua preocupação em padronizar o conteúdo dos livros da série. Assim, estes mesmos apêndices são incluídos nos outros componentes da série SOFTWARE. A composição destes apêndices é:

APÊNDICE A – "Como Utilizar os Programas deste Livro".

É apresentada a padronização dos programas componentes do livro – especificamente a documentação que é subdividida em "Ficha", e três manuais.

- FICHA: Uma descrição das características do programa – Nome do autor, versão, data, linguagens utilizadas, microcomputador e compatíveis, recursos necessários (memória RAM, "joystick", impressora, ...), instruções/comandos especiais (SLOW, LLIST, DLOAD, ...) e observações.
- MANUAL DE UTILIZAÇÃO: Apresenta o enunciado do problema e a descrição do uso do programa.
- MANUAL DE OPERAÇÃO: Como operar o programa – como instalá-lo, processamento normal e exceções e erros.
- MANUAL TÉCNICO: A solução do problema.

APÊNDICE B – "Carregamento de Programas a partir de Fitas Cassete".

Seguem instruções de como ligar e acoplar o micro e gravador, ajustes nos controles do gravador e outras informações necessárias para se carregar um programa com sucesso.

ATENÇÃO:
Os programas listados no livro podem ser solicitados em Fita K-7 à EDITORA CAMPUS LTDA.

APÊNDICE C – "Armazenamento de Programas em Fita Cassete".

Os cuidados que devem ser tomados para armazenar seus programas em fita cassete.

APÊNDICE D – "Cuidados com gravadores e Fitas Cassete".

Como escolher um gravador de boa qualidade e recursos necessários e cuidados a serem tomados quanto a operação e limpeza.

Escolha de fita cassete – qualidade, tipo (normal ou baixo ruído), cuidados com seu uso e armazenamento.

Ao estudar estes apêndices, o leitor irá tirar o maior proveito desse livro e também dos outros componentes da série por causa da padronização exigida pelo coordenador.

Chegando, finalmente, ao livro em questão, o autor informa ao leitor que a obra não é de cálculo numérico, mas de aplicação do mesmo. Outro ponto destacado é que, embora os programas fossem desenvolvidos em um TK82-C com expansão de 16K, houve uma preocupação constante de utilizar ao máximo o BASIC padrão para facilitar a compatibilização entre outros equipamentos.

O conteúdo do livro é composto de:

Capítulo 1: Matrizes: Descrição de matrizes em geral – retangular, quadrada, linha, coluna, determinante: Operações básicas – adição, subtração, multiplicação, transposição, inversão (Programa POM1). Operações entre matrizes e escalares (POE1). Operação com linhas e colunas (Programa POL1). Outras operações envolvendo matrizes (Programa POO1).

Capítulo 2: Raízes de Equações: Funções, polinômios, limite de iterações, convergência, raízes reais e complexas, derivada de uma função, métodos Newton-Raphson, Regula-Falsi, Bissecção (rotinas RRE1NER, RRE2REF, RRE3BIS no programa PRE1).

Encontro de todas as raízes de um polinômio – programa PTR1.

Capítulo 3 a 6: Seguem os capítulos com os tópicos:

- **Capítulo 3:** Resolução de Sistemas de Equações Lineares.
- **Capítulo 4:** Polinômios.
- **Capítulo 5:** Integração Numérica.
- **Capítulo 6:** Estatística.

Em suma, este livro é uma excelente fonte de programas na utilização de métodos numéricos num microcomputador. A apresentação dos algoritmos, fluxogramas, programas e manuais é bem clara e há uma preocupação excepcional em manter uma padronização, que suponho será mantida nas outras obras desta série SOFTWARE. Os programas como estão podem ser utilizados imediatamente nos micros da linha Sinclair, e não deverá haver grandes dificuldades em transformá-los para outros tipos de micro.

EDIÇÕES AVULSAS

Complete a sua coleção de MICROBITS solicitando os números que lhe faltam.
Verifique a matéria publicada nas edições anteriores:

Ano I – Nº 1
- Introdução ao CLUBE NACIONAL DOS TK/NE/SINCLAIR
- Programa "PLOTAR"

Ano I – Nº 2
- Linguagem de máquina – Onde e Como?
- "Benchmark" – BASIC comparada a linguagem de máquina
- Como Criar um Cadastro (Parte 1)
- Programa "CORRIDA DE CARROS"

Ano I – Nº 3
- Programa "CONTAS A PAGAR"
- Como Proteger Seus Programas
- Conserto de um Programa Defeituoso
- Teclado Musical
- Conversão Hexadecimal-Decimal
- Medidor de Intensidade de Saída do Gravador para "LOAD"

Ano I – Nº 4
- Armazenar a Imagem Numa Variável
- O TK85 da Microdigital – Vale a Pena?
- Testando o TIG-LOADER
- Programa "SENHA"
- Programa "VALOR PRESENTE"

Ano I – Nº 5
- BENCHTEST: ANÁLISE DO RINGO
- Maior Resolução Gráfica
- Alta Resolução no Vídeo
- VU Incrementado
- Codificador de Texto
- Submarino Nuclear
- Curvas Paramétricas

Ano I – Nº 6
- Como Criar Caracteres Especiais no Seu Micro – Parte I
- Linguagem de Máquina: Como Usar a Tela
- Benchtest: CMS/ZX da Polimicro
- Como Criar um Cadastro (Final)
- Sistema de Equações

IMPORTANTE

Qualquer dos seis números publicados no Ano I pode ser obtido mediante envio de cheque nominal diretamente a:

MICRO BITS INFORMÁTICA LTDA.
CAIXA POSTAL 12.464
CEP 04798 – SÃO PAULO – SP

Ano II – Nº 1
- Como Criar Caracteres Especiais no Seu Micro – Parte II
- Salvando e Restaurando Telas
- Benchtest: Teclado da Speed
- Consumo
- Crie uma Linha REM
- Controle de Gravador K7
- Benchtest: Gluckstick Digital
- PERT
- Jogo de Inversão
- Caracteres de Vídeo
- Desenhe na Tela
- Interface para Telegrafica (C.W.)
- Números Primos

Ano II – Nº 2
- A Grande Arma
- Como Criar Caracteres Especiais no Seu Micro – Parte III
- Teclados de Membrana
- FORCA em Linguagem de Máquina
- Além do BASIC
- Programa PIRATAS
- Avenida II

IMPORTANTE

Os números publicados a partir do Ano II nº 1 podem ser obtidos mediante envio de cheque nominal diretamente a:

Editora Campus Ltda.
Rua Barão de Itapagipe, 55
CEP 20261 – Rio de Janeiro – RJ
Tel.: (021) 284 8443

- Números antigos serão vendidos ao preço de capa do último número lançado.
- Este valor já inclui as despesas de correio.

EDITORA CAMPUS LTDA.

NO UNIVERSO DA COMPUTAÇÃO OS LANÇAMENTOS

CAMPUS

OCUPAM TODOS OS ESPAÇOS

1 **Besant, C. B.**
CAD/CAM – Proj. e Fabricação com o Aux. do Computador
Cr$ 46.200

2 **Botelho, A. J. L.**
Desafio. Os Mais Excitantes Jogos em BASIC
Cr$ 37.900

3 **Chance, D.**
30 Programas em Basic para Computadores Pessoais
Cr$ 39.900

4 **Longworth, G.**
Padrões em Programação: Métodos e Procedimentos
Cr$ 48.000

5 **Chmura, L. J. e Ledgard, H.**
COBOL com Estilo
Cr$ 24.900

6 **Karsten, L.**
Programas Administrativos em BASIC SINCLAIR
Cr$ 24.900

7 **Loiola, C. R. A.**
Rotinas Matemáticas em BASIC para Micros
Cr$ 62.800

8 **Hergert, D.**
BASIC para Aplicações Comerciais
Cr$ 20.000

9 **Pereira Fº, J. C.**
BASIC Básico (5ª Edição Ampliada e Revisada)
Cr$ 27.800

10 **Meili, E.**
APLICALC, um Software Educacional, Pessoal e Profissional em BASIC
Cr$ 23.800

11 **Christmann, R. U.**
Basic Sinclair
Cr$ 19.900

12 **Schware, R. e Trembour, A.**
Tudo Sobre 1–2–3
Cr$ 22.000

13 **Garbin, E. A.**
Introdução ao VISICALC (2ª Edição)
Cr$ 24.600

14 **Hughes, C.**
Primeiros Passos com o seu SINCLAIR (TKs)
Cr$ 18.800

15 **Christian, K.**
Sistema Operacional UNIX
Cr$ 54.900

EDITORA CAMPUS LTDA.
Rua Barão de Itapagipe 55 – 20261 – Rio de Janeiro – RJ – Tel.: 284 8443

- Desejo receber as publicações indicadas abaixo

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

☐ Mediante envio de cheque nominal à Editora Campus Ltda. (despesas postais por conta da Editora). Nº ..... Banco .....

☐ Mediante pagamento contra recebimento dos livros pelo reembolso postal, acrescido das despesas postais.

Nome: .....
Endereço: .....
Cidade: ..... *CEP:* ..... *Estado:* .....

IMPORTANTE: Preços válidos por tempo limitado.

Otavio Studart